CS 6905 – Fall 2017

Software Engineering

Online Voting System

Submitted by:

Group: # 4

Aafia Jabeen – 201790648

Aswini Naveen – 201692818

Haris Khan – 201454147

Due Date:

1 December 2017

Contract

Software Development Agreement

Our team has been contracted to design a system to develop an online voting system that will provide support for election officials to conduct an election.

Requirements

The vision of this project is to design generic Online-Voting system that will provide support for election officials to conduct an election (e.g. parliamentary, student union, sport club). This system design support popular election systems primarily winner takes all or proportional. If in case election officials decide to choose proportional election model then there is a threshold percentage votes to be achieved for a party to gain membership in a parliament. More than a one round of elections is possible. Additionally, the system shall provide directions to voting stations. For winner takes all, the election official can also select if the voter can select preference for candidates. After election the least popular candidate is dropped and votes re-assigned according to second choices, and so on, until one candidate has a majority of ballots. The election official will also be able to select districts participating in election and will be able to update the candidates.

We have agreed that the program will be developed in NodeJS, the frontend in Angular and database would be MongoDB.

Developers

Aafia Jabeen

Ashwini Iyer

Haris Khan

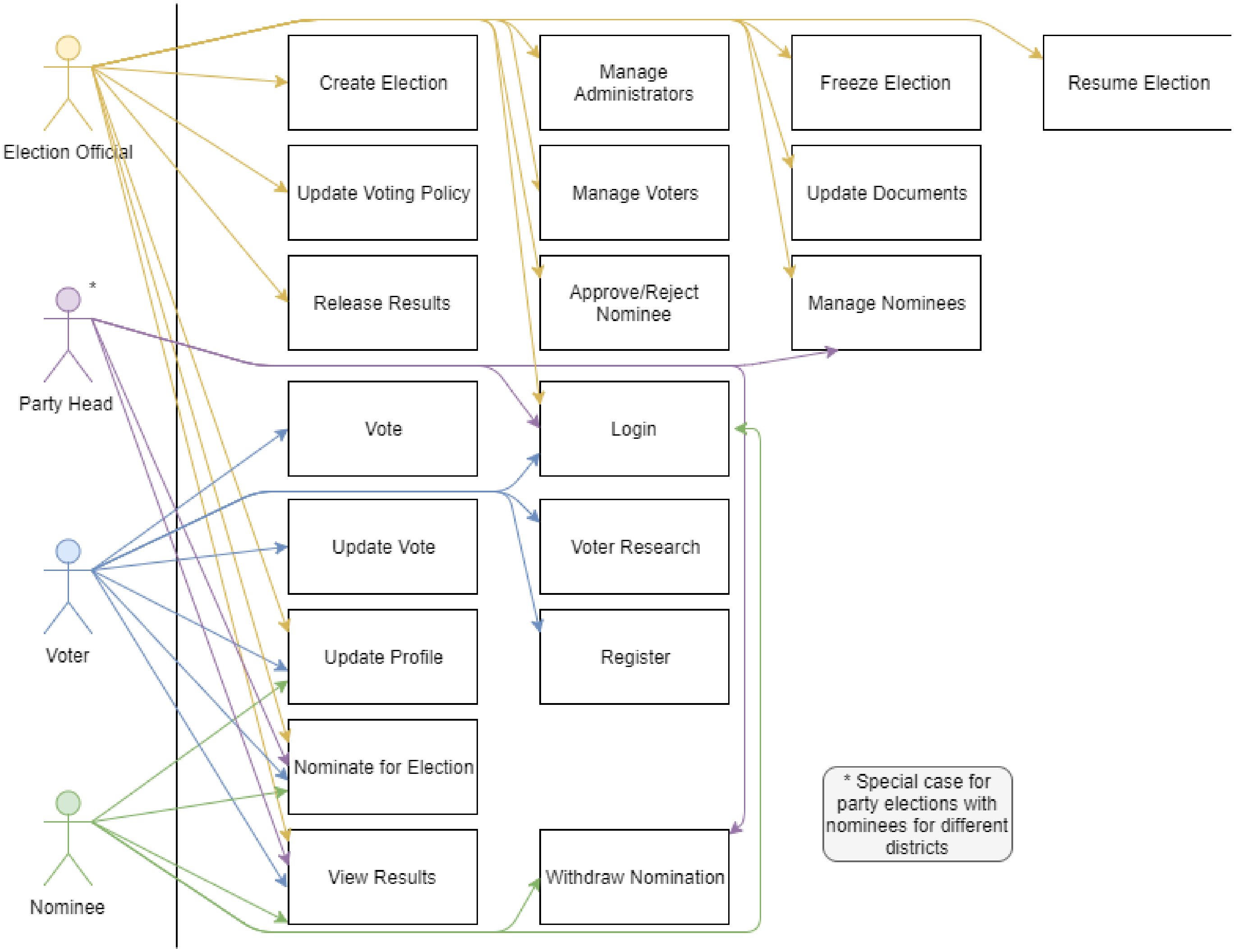
Client

Dr. Adrian Fiech

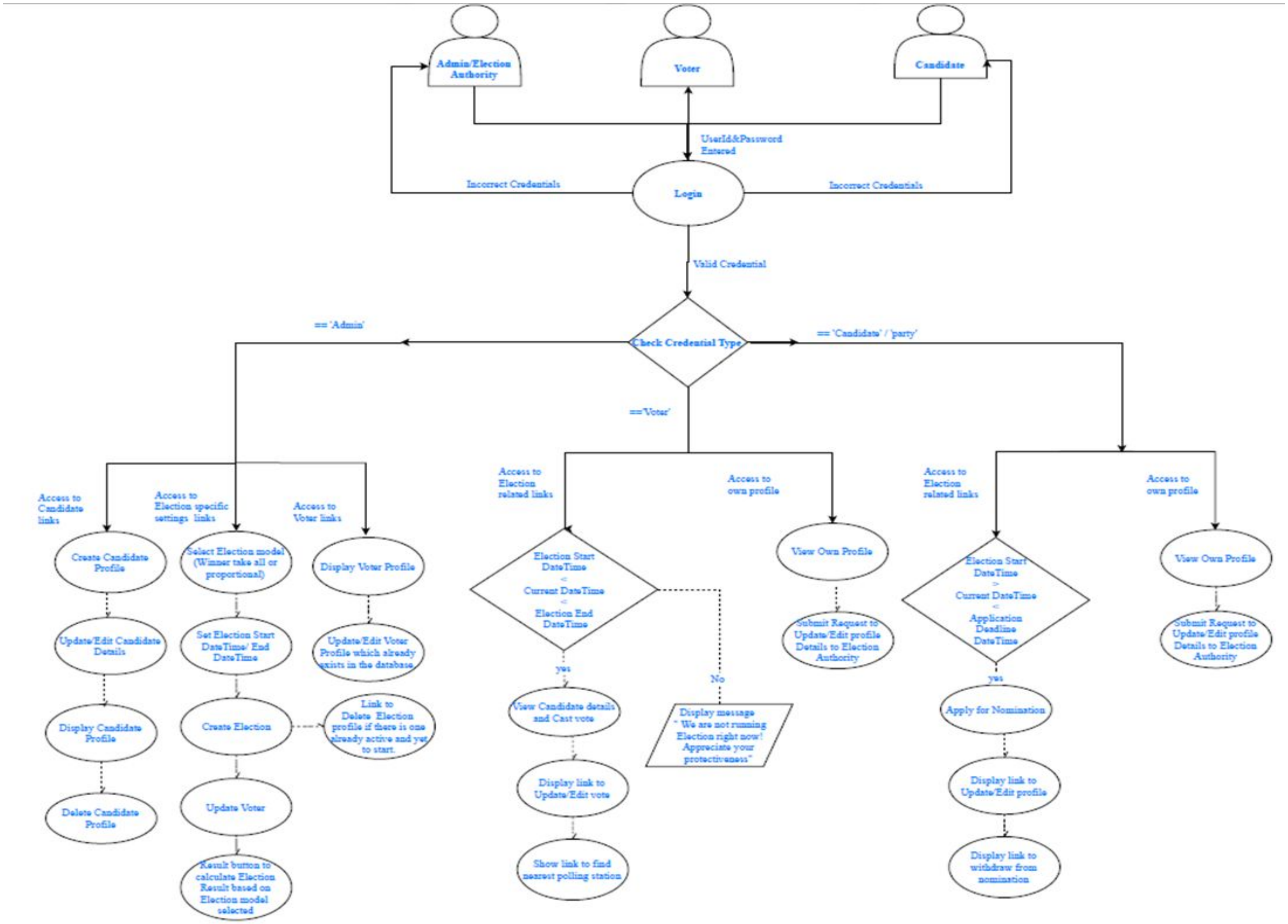
Use Case Diagram

Use Cases Described Below:

1. **Create election**
2. **Vote**
3. **Nominate for Election**
4. **Voter Research**
5. **Release results**
6. **Update Vote**
7. **Freeze Election**
8. **Update Documents**
9. **Update Voting Policy**
10. **Nominee Update Profile**



Alternate Use Case Diagram – System Level (Basic)


Admin/Election Authority
Voter
Candidate
UserId&Password Entered
Incorrect Credentials
Login
Incorrect Credentials
Valid Credential
Check Credential Type
== 'Admin'
== 'Candidate' / 'party'
=='Voter'
Access to Candidate links
Access to Election specific settings links
Access to Voter links
Create Candidate Profile
Update/Edit Candidate Details
Display Candidate Profile
Delete Candidate Profile
Select Election model (Winner take all or proportional)
Set Election Start DateTime/ End DateTime
Create Election
Update Voter
Result button to calculate Election Result based on Election model selected
Display Voter Profile
Update/Edit Voter Profile which already exists in the database
Link to Delete Election profile if there is one already active and yet to start.
Access to Election related links
Access to own profile
Election Start DateTime < Current DateTime < Election End DateTime
yes
No
View Candidate details and Cast vote
Display link to Update/Edit vote
Show link to find nearest polling station
Display message " We are not running Election right now! Appreciate your protectiveness"
View Own Profile
Submit Request to Update/Edit profile Details to Election Authority
Access to Election related links
Access to own profile
Election Start DateTime > Current DateTime < Application Deadline DateTime
yes
Apply for Nomination
Display link to Update/Edit profile
Display link to withdraw from nomination
View Own Profile
Submit Request to Update/Edit profile Details to Election Authority

Use Cases

Use Case # 1 - Create Election

Name:

Create Election

Description:

This use case will allow the election official (organizer) to conduct an election. The election official will create the kind of election to be conducted (eg: parliamentary, student union, etc) and will also update the details of various districts (where the elections to be held) and the nominees. Upon request from the nominee or the voter, the use case will allow the election official to make changes in the personal details of the nominee or voter after proper verification. The election official prepare electoral roles and will update the voter's list from time to time. The schedule of election for filing the nominations are also issued. The election official will set up the rules for voting like the eligibility criteria, documents to be submitted and the verification required to apply for nomination, time period for a particular election and the number of rounds to be conducted. It lays down guidelines for the conduct of political parties and candidates during an election period. After the elections are conducted the election official will also be responsible for the release of results.

Actor:

- **Election official**

Basic Flow:

1. **Election official signs into their account.**
2. **Selects create election**
3. **Update the election settings for the kind of elections to be held**
4. **Set the time for when the election starts**
5. **Set the time for when the election ends**
6. **Select winning strategy (e.g. winner takes all, proportional)**
7. **Select number of districts**
8. **Select predefined candidates or nomination based**
9. **Select the users who can vote (database, csv file, regular expression)**
10. **Add conditions for the user object that can vote (e.g. if only graduate students can vote or people can vote for their district only)**
11. **Select create**

Alternate Flow:

6.
   a) If (winner takes all) then select single vote or preferential vote

8.
   a) If predefined selected then add candidates for districts
   b) If nomination selected then
      i) Select either nominations has to be approved by an official or not
      ii) Add regular expression for users who can nominate (optional)

9.
   a) If database then add the credentials and API URL for the database and the query for the database
   b) If csv file then either add the URL and credentials to access it or upload the file
   c) If regular expression then add the regular expression for the email address

11.
   a) Select save as draft.

Preconditions:

- Election official is registered
- Election official is verified

Postconditions:

- Election to be held is created (if not saved as draft)
- Personal details of the nominee/voter are updated (if applicable)
- Cancels the nomination for a nominee (if applicable)

Use Case # 2 - Vote

Name:

Vote

Description:

The purpose of this use case is to allow a voter to cast a vote. The voter will have privileges like casting a vote, modifying it later and also to view the results. All eligible voters (residents, students, specific group of people) will be pre-registered and given access to the application. Voter can change personal details which will be validated by the system and stored in the database.

Actor:

- **Voter**

Basic Flow:

1. **Voter signs into their account.**
2. **Selects vote to cast a vote**
3. **View for which candidate he has voted.**
4. **Selects update vote (if applicable).**
5. **View the results after the elections are over.**

Alternate Flow:

1. **Voter signs into their account.**
2. **Cannot cast a vote (deadline has passed for voting)**
3. **Voter is notified by the system**
4. **View the results**

Preconditions:

- **Voter is registered**

Postconditions:

- **Able to cast a vote**
- **Voting results can be displayed**
- **Personal details are updated (if applicable)**

Use Case # 3 - Nominate for Election

Name:

Nominate for Election

Description:

This use case goes through the process of self nomination. If in case the candidate is new he /she can register their own profile and get an username/password assigned. If the candidate belongs to a specific party provided the party or the candidate already has existing account then they can login to the account and submit a new nominee application or update their existing profile (if required) or withdraw from nomination.

Actor:

- **Nominee**

Basic Flow:

1. **User signs into their account.**
2. **Selects the election.**
3. **Selects nominate.**
4. **Selects self nominate or nominate somebody else.**
5. **Enter the details for the nominee.**
6. **Save**

Preconditions:

- **User is registered.**

Postconditions:

- **New nominees are sent for approval or are registered depending on voting policy**

Use Case # 4 - Voter Research

Name:

Voter Research

Description:

This use case will allow the voter to perform some research on the website by selecting a research task from the list. A voter can select the category of election and the type of information he is looking for: election laws, nominees list, the policies and objectives of the political parties , etc. The voter is then presented with the appropriate type of documents by the system stored in the database. He can choose to view any number of required documents from a menu of options for the selected category of research. A voter can also search for the document he is looking for through a 'search' tab. The system will extract the required information from the database.

Actor:

- **Voter**

Basic Flow:

1. **Voter signs into their account.**
2. **Selects the category of election for which the information is required**
3. **Selects the type of information from the list**
4. **Selects the appropriate document**
5. **Selects 'stop viewing' if he wants to stop looking for information.**
6. **Voter is asked if he needs any other information from the system and if he got all the information he was looking for**
7. **If yes, voter will be asked if he would like to perform any other task in the session**
8. **If no, voter will be redirected to the research task list.**

Use Case # 5 - Release Results

Name:

Release results

Description:

This use case declares the winning candidate or the political party after the elections. It will determine the number of political parties or independent candidates participated in the elections, count the number of votes for each candidate or political party, decide the candidate with the maximum number of votes and declare the winner. It will also determine the voting difference for which a particular candidate has won the election and release the result along with the winning candidate details.

Actor:

- **Election official**
- **Administrator**

Basic Flow:

1. **Election official/Administrator signs into their account.**
2. **Select the election.**
3. **Selects the category of election.**
4. **Checks the number of political parties / candidates participated in the election.**
5. **View the candidate with the maximum number of votes.**
6. **Release the results for winning candidate or political party along with candidate details.**

Use Case # 6 - Update Vote

Name:
Update Vote

Description:

This use case allows users to change vote if they had voted already. Once the user is logged in the users are presented with list of election's those are created and that those they are eligible to vote. If the user selects the election that he has not voted and the deadline to vote is still open then he gets redirected to cast vote page with a message indicating he will be eligible to change after he/she had completed voting. If the user selects the election that he had already voted and deadline to vote has already passed then then he will be displayed with the details of his choice. If the user selects the election that he had already voted and the deadline to vote is still open then he is he will be eligible to change his vote.

Actor:
Voter

Basic Flow:

1. **Voter is Registered and logged in.**
2. **Provided with list of Elections conducted so far and that he/she was eligible to vote.**
3. **Chooses the election to change vote.**
4. **After changing, selects Yes or No for final confirmation.**
5. **If Yes selected the latest changes are saved.**
6. **If No, then old value is retained.**

Use Case # 7 - Freeze Election

Name:

Freeze Election

Description:

The Election Official or the administrator will be able to freeze the election so that nobody can vote. This would be a special case scenario when the election has to be halted, it can be resumed later on.

Actor:

- **Election official**
- **Administrator**

Basic Flow:

1. **User is authenticated as election official or administrator**
2. **User selects the election to be frozen.**
3. **User selects freeze.**
4. **User authenticates again.**

Use Case # 8 - Update Documents

Name:

Update Documents

Description:

The Election Official or the Administrator have the privilege to edit/update the document/profile of candidate or the voter.

Actor:

- **Election Official**
- **Administrator**

Basic Flow:

1. **The Election Official or the Administrator is signed in.**
2. **Selects Update Documents.**
3. **Choose type of user Voter or Candidate.**
4. **Provides the unique UserId (either Voter_id/Candidate_Id/*Party_Id).**
5. **Clicks search to locate the profile.**
6. **Once profile is loaded edits the document.**
7. **Clicks on Save changes.**

Use Case # 9 - Update Voting Policy

Name:

Update Voting Policy

Description:

This use case allows the Election official or the administrator to update the voting policy for a particular election. The voting policy will be set by the election official at the time of creating the election and can be updated later if required. In order to make changes in the policy, the admin/election official is required to select the category of election for which the voting policy needs to be updated. The admin/election official is required to enter valid input data example date and time in the system. If the data is invalid, he will be notified by the system through an error message and will be asked to re-enter the information or cancel the task. After the voting policies are updated, submit can be used to save the data.

Actors

- **Admin**
- **Election Official**

Basic Flow:

1. **Admin is logged in.**
2. **Selects the category of election for which the voting policy needs to be updated.**
3. **Make the changes. Enter the valid data.**
4. **In order to save the data, click on submit.**
5. **The admin will be asked to confirm the changes made.**
6. **If yes is selected, the changes are saved and the voting policy is updated.**

Alternate Flow:

1. **Admin is logged in.**
2. **Selects the category of election for which the voting policy needs to be updated.**
3. **Make the changes. Enter the invalid date.**
4. **An error message generates and admin is asked to enter the valid date or cancel the task.**
5. **Admin cancels the task and close the session.**

Preconditions:

- **Admin is registered**
- **Admin is verified**
- **The voting policy is already set for that particular election by the election official**

Postcondition:

- **The voting policy is updated.**

Use Case # 10 - Nominee Update Profile

Name:

UpdateProfile - Nominee

Description:

This use case enables the registered nominees to make changes in their personal information by providing the appropriate data with the supported documents. The nominee selects the type of information he needs to update example: personal information, background details. After the data is entered, the nominee will be asked to upload the supported documents. The documents will then be verified by the election official before the final changes reflects in the system. Till the election official approves the changes, the request will be in pending state. After making the changes and uploading the documents, the nominee can save the changes and close the session.

Actors

- **Nominee**

Basic Flow:

1. **Nominee is logged in.**
2. **Selects the category of information he would like to update.**
3. **Enter the required and valid data.**
4. **Upload the supported documents.**
5. **Save the changes through the submit option.**
6. **Nominee will be asked by the system to confirm the changes made.**
7. **Select yes and close the session.**

Alternate Flow:

1. **Nominee is logged in.**
2. **Selects the category of information he would like to update.**
3. **Enter the required and valid data.**
4. **Upload the supported documents**
5. **The task failed as the system fails to accept the uploaded document.**
6. **Nominee is asked to upload the document again.**
7. **The task failed again and after a number of failed attempts the task is cancelled.**
8. **Nominee is asked if he likes to perform another task.**

9. **Selects No and close the session.**

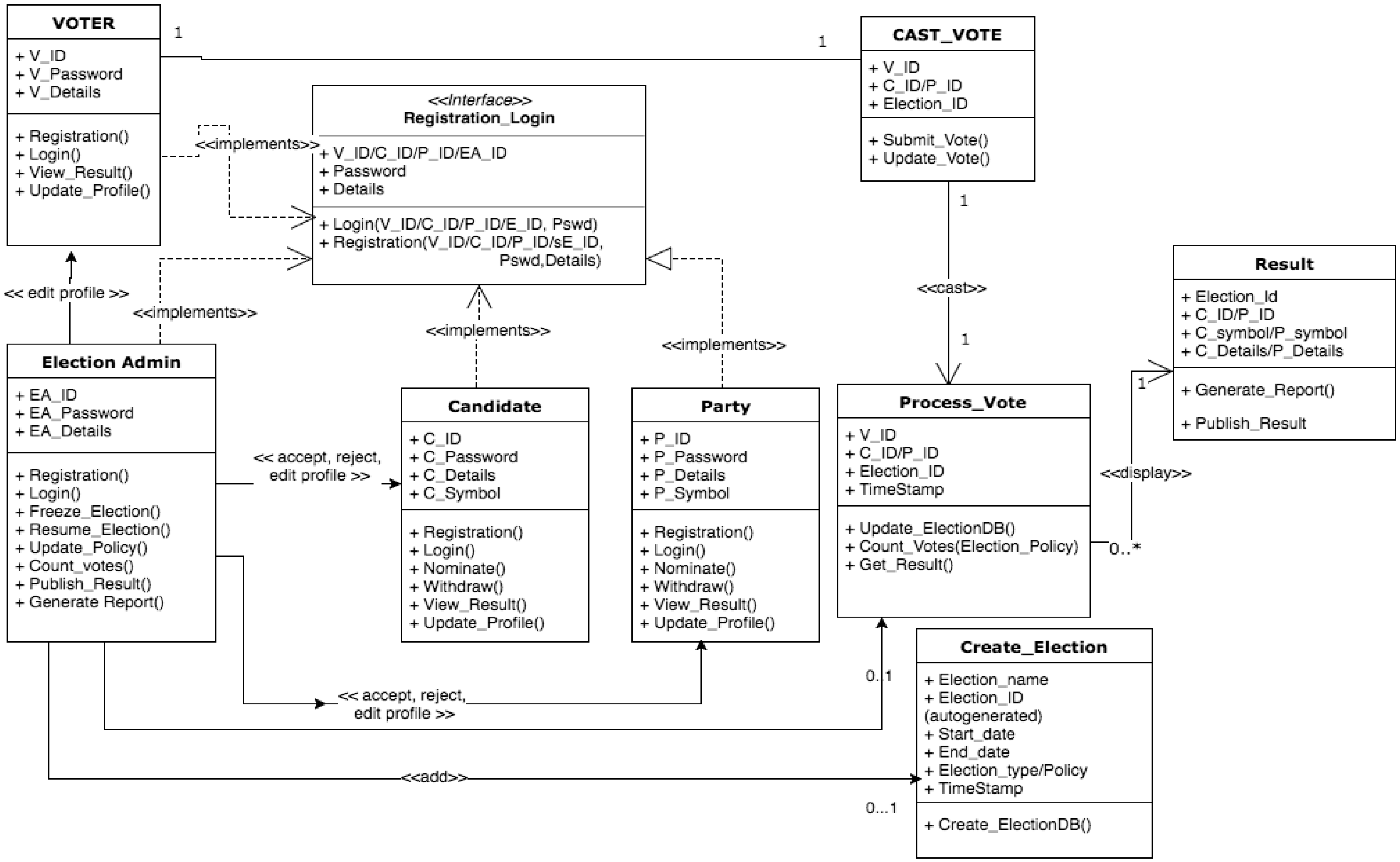
Preconditions:

- **Nominee is registered**

Postcondition:

- **Personal details of the nominee is updated.**

Domain Modelling



Design Goals

Below mentioned design goals were considered while architecting system design for Online Voting System.

Reliability
The system is reliable and designed to cater up all the basic functionalities of an Online Voting System.

Response Time
The Response Time is very minimal as we use MongoDB, Express, Angular and Node.js(MEAN stack) and the throughput rate is very high compared to other MVC frameworks.

Robustness
The system is designed to survive invalid user input. For example: if a voter wants to cast a vote for an election after the deadline, the system will notify the user that the deadline has already passed and the voting is closed.

Security
The system is secure and not vulnerable to malicious attacks. The candidate, voter details and the rules of the elections should be protected at all times and should not be accessible to any external entity.

Portability
The system should be portable. It is developed using HTML and node.js and hence can be adapted to any operating systems and platforms.

Availability
The system should be available most of the time; it can be used to accomplish normal tasks. The downtime of the server should be minimum or almost never. During downtime, new registration of the nominee will not be done. Only the registered nominees and the voters can be viewed. The system will be in a view mode only.

Usability
The system should be user-friendly so that an individual with not much basic web browsing skills can use the system with ease.

Scalability

With the increase in the number of voters and candidates for a particular election, the system should be able to handle the data accordingly and should function properly. As we have used MVC architecture to design the system, the HTML pages will get connected to the database which will handle the traffic well.

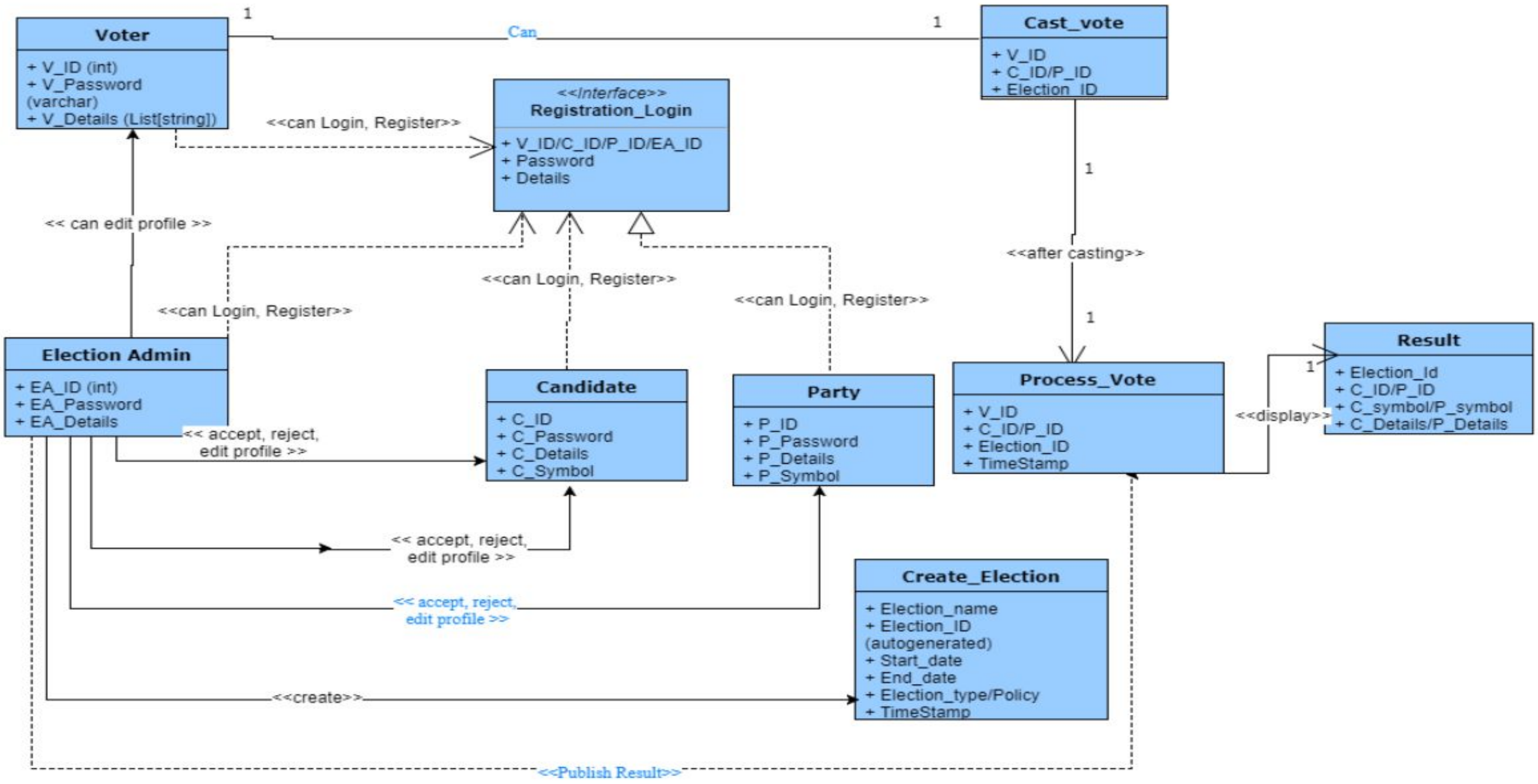
Performance

The information/data should be stored in a properly designed database. The frontend will be designed using node.js to serve the web pages and the database using MongoDB. Both the technologies are compatible with each other; hence the connectivity and the data exchange rate will be done faster. It's important to test the system thoroughly for bugs before the deployment.

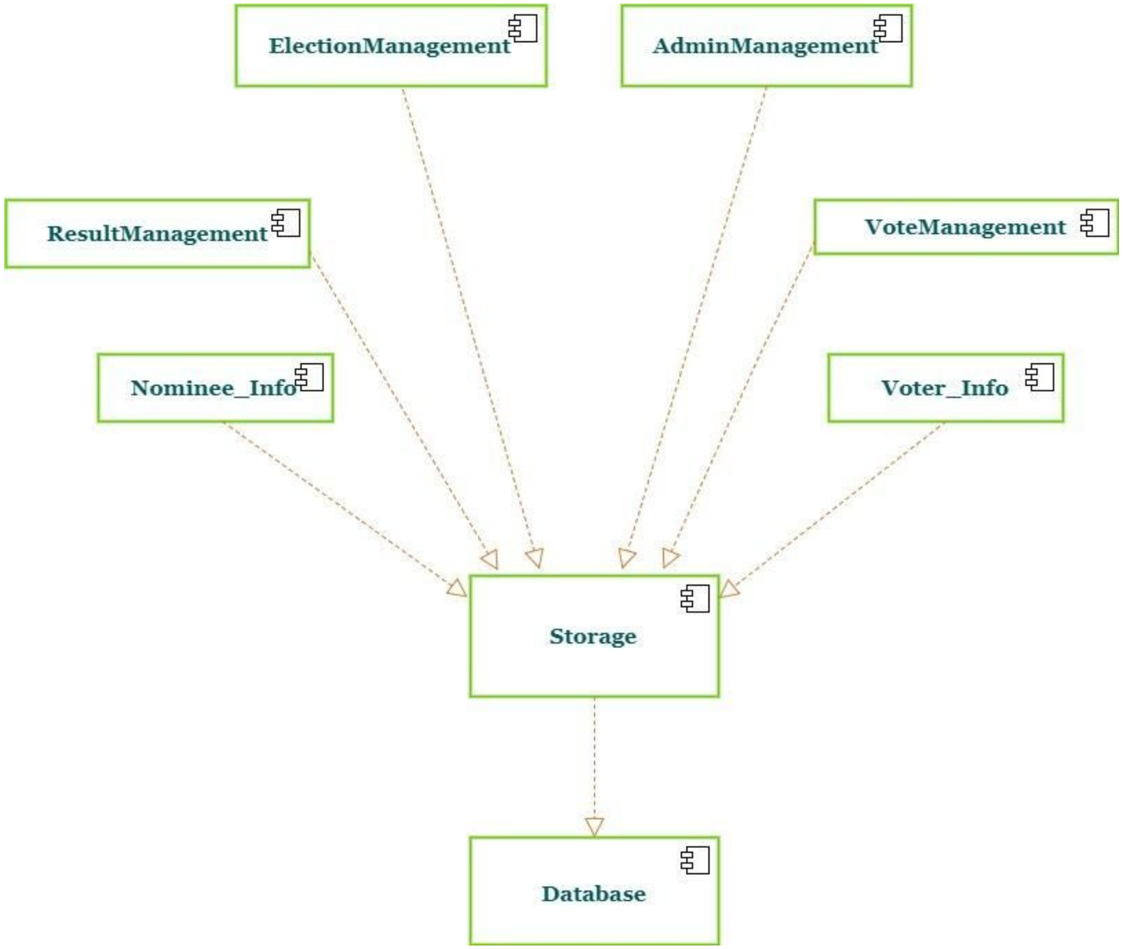
Utility

The system should support the work of Election Official and should be able to conduct the elections as per the rules the defined. The voters can cast the votes and nominees can nominate themselves. The evaluation of the results and declaring the winner should be accurate.

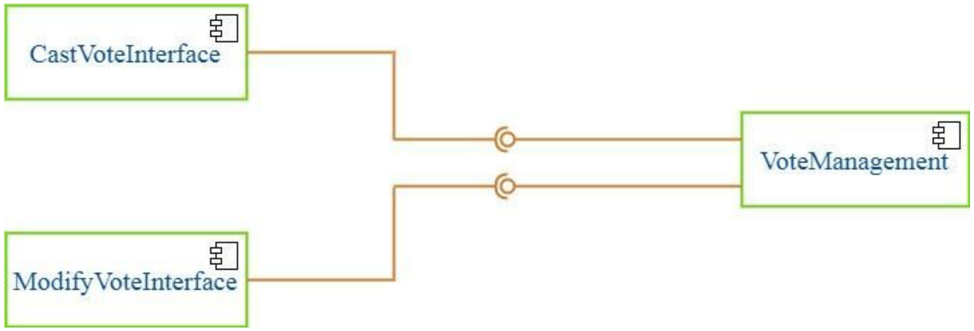
Object Modelling:



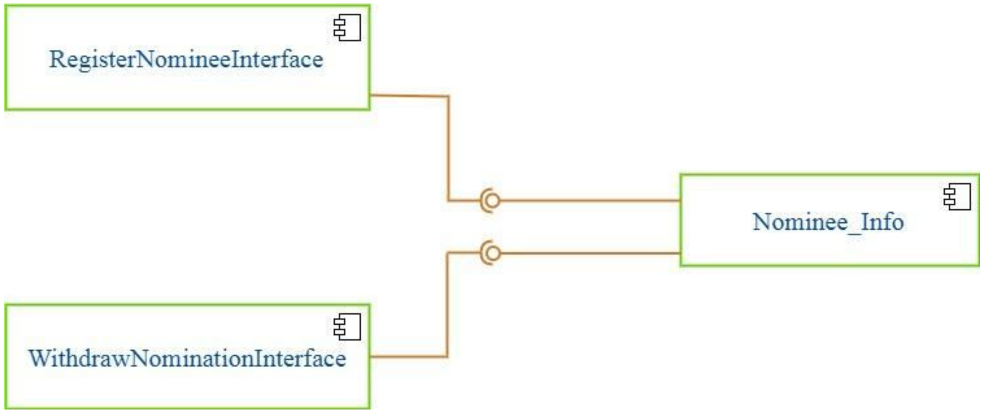
System Decomposition



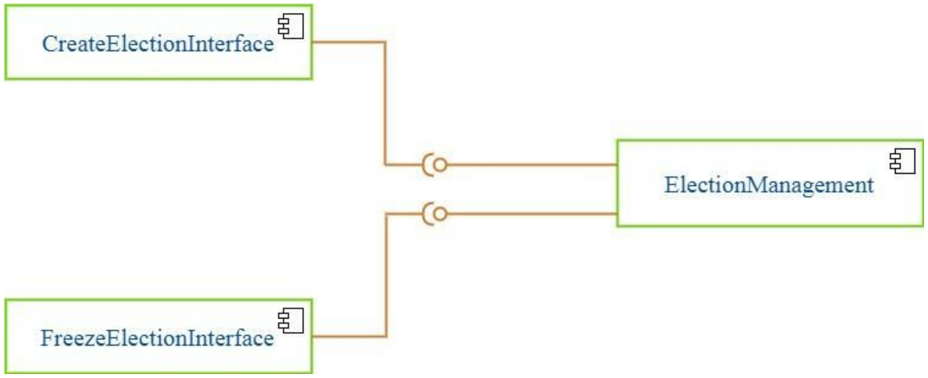
Services provided by the subsystems (UML component diagram, ball-and-socket notation depicting provided and required interfaces)



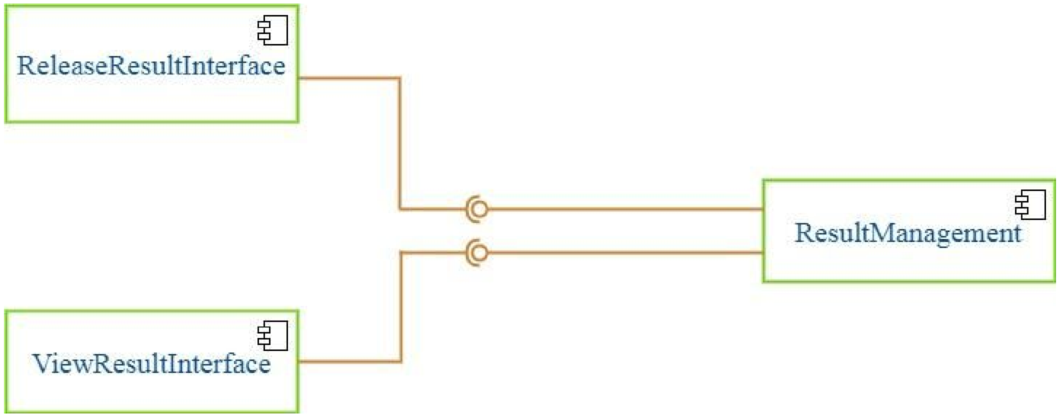
The VoteManagement subsystem provides services to the other modules. A voter can cast a vote by using the CastVoteInterface during the time elections. After the voting is done and the deadline has not passed for a particular election, ModifyVoteInterface enables the voter to update his vote.



RegisterNomineeInterface and WithdrawNominationInterface require services from the Nominee_Info subsystem in order to maintain the list of nominees for a particular election.



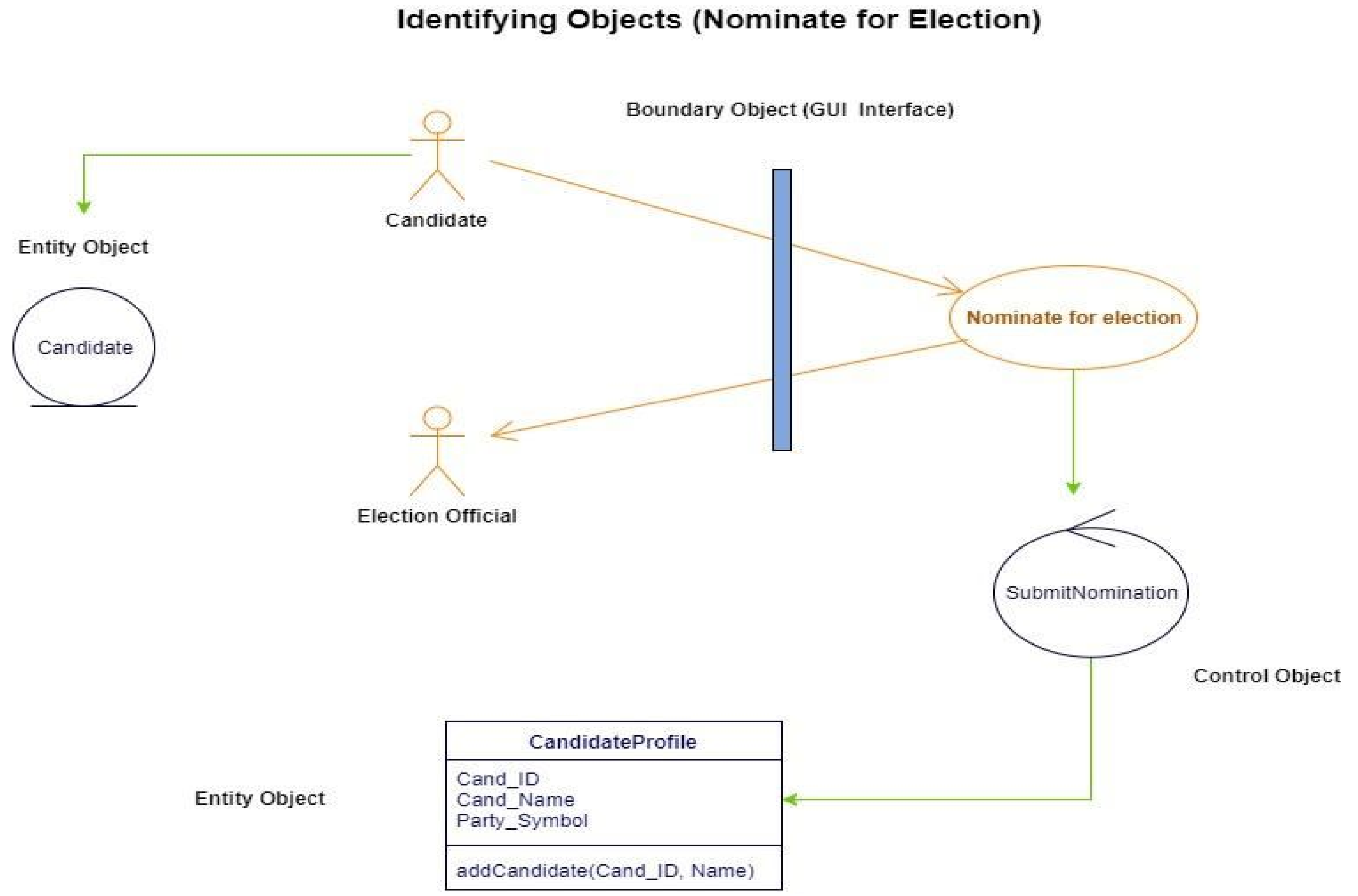
CreateElectionInterface and FreezeElectionInterface requires services from the ElectionManagement subsystem for the proper conduct of elections.



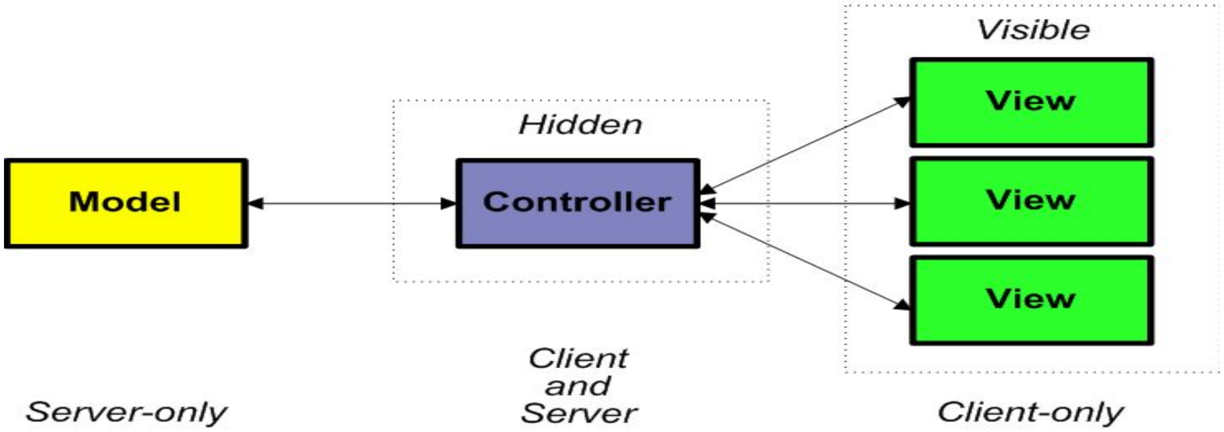
ReleaseResultInterface and ViewResultInterface requires services from the ResultManagement subsystem for declaring the winner after the elections are over.

Entity, Boundary And Control Objects

The frontend are the boundary objects, the frontend will be developed in Angular and will have MVC (Model View Controller) architecture type of it's own. So all the forms and webpages the user will see will be boundary objects. The control elements will contain in NodeJS and will have MV*(Model View) architecture type. So all the classes in domain modelling are actually controller objects. All these classes will be entities as well which will be stored in the database.



Logical Architecture: Model View Controller (MVC)

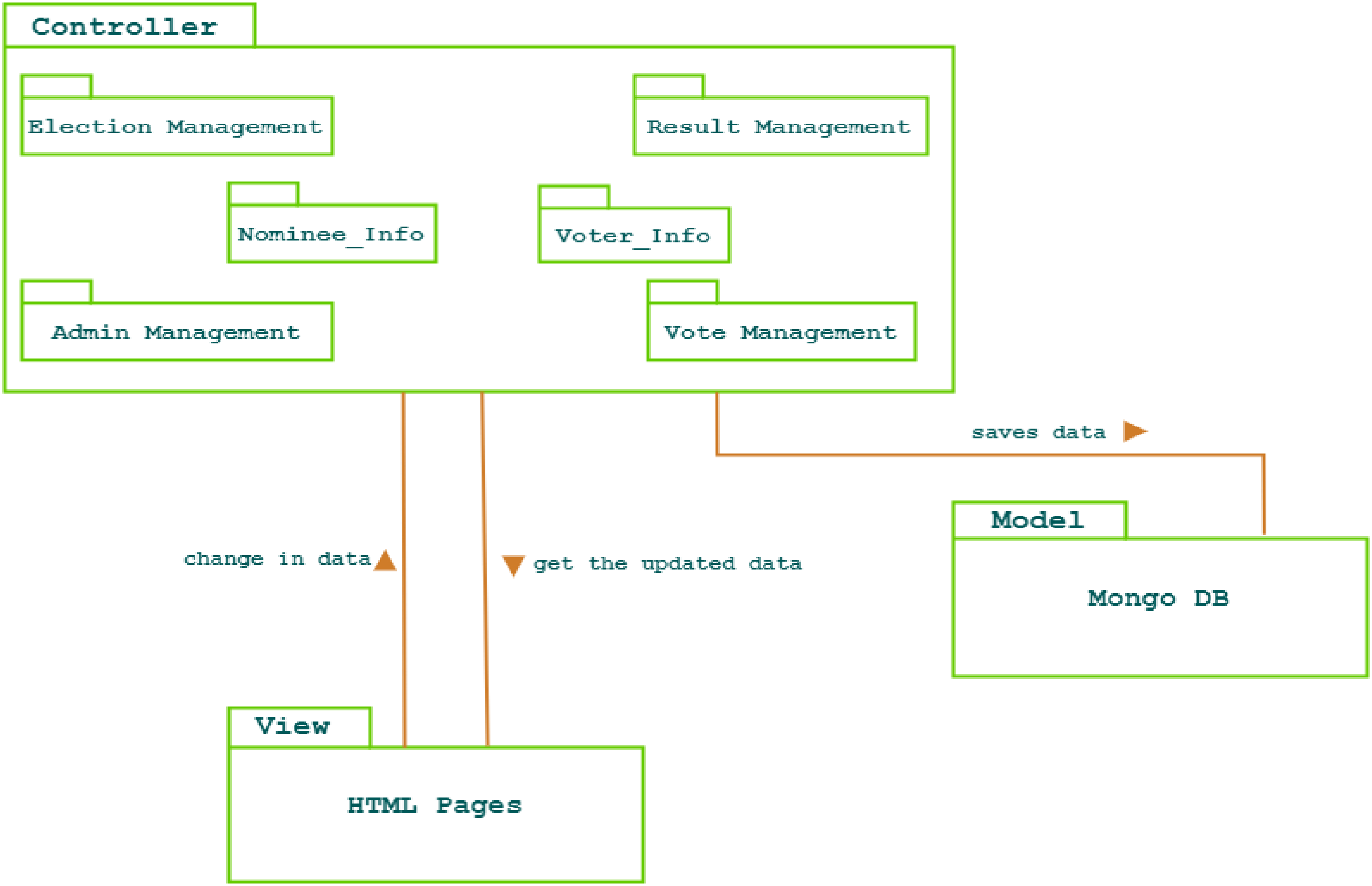


We are going to use the Model View Controller architecture for our application. Model is the central component of the architecture and it handles the access to the database server (Mongo DB). Controller acts as an intermediate and accepts the input from the user and convert it to commands for the model or view. We will be using Node.js as a controller in our application. View handles the entire HTML pages. It is the output representation of the data that the user interacts with.

Model: Mongo DB

View: HTML in Angular (MVC framework on frontend)

Controller: Node.js (MV*)


Controller
Election Management
Result Management
Nominee_Info
Voter_Info
Admin Management
Vote Management
saves data
change in data
get the updated data
Model
Mongo DB
View
HTML Pages

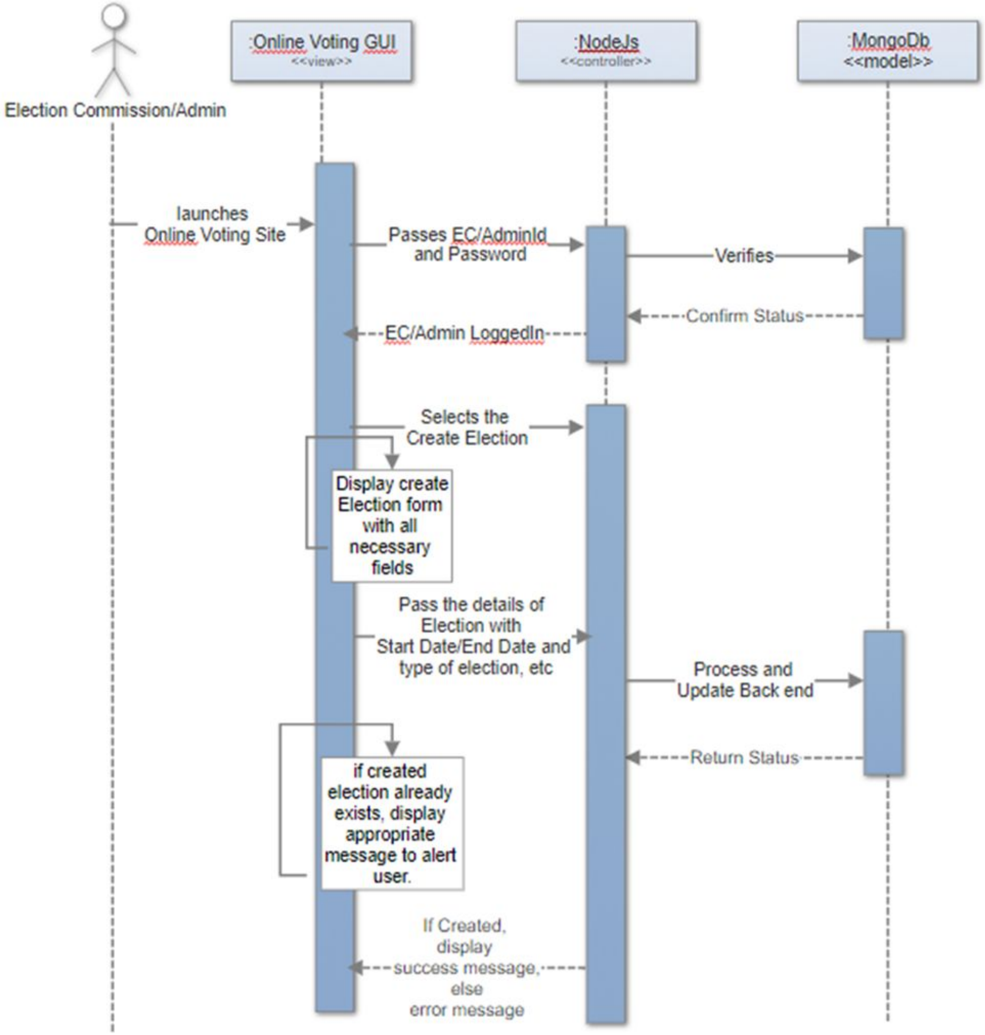
Sequence Diagrams

A sequence diagram is an interaction diagram that shows how objects interact with one another and in what order.

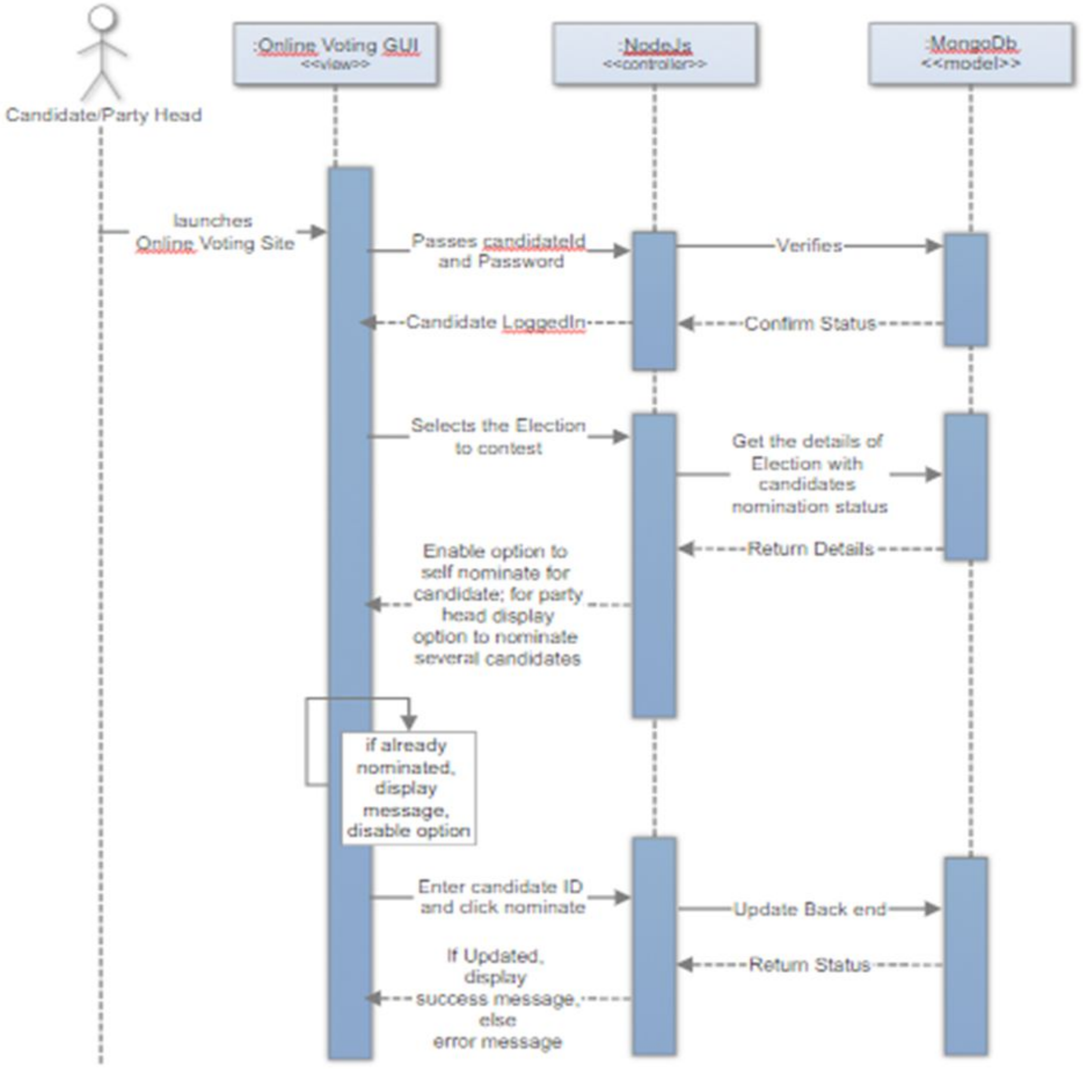
Here we have created sequence diagrams of *three essential use cases*:

- **Create Election**
- **Candidate Nomination**
- **Caste Vote**

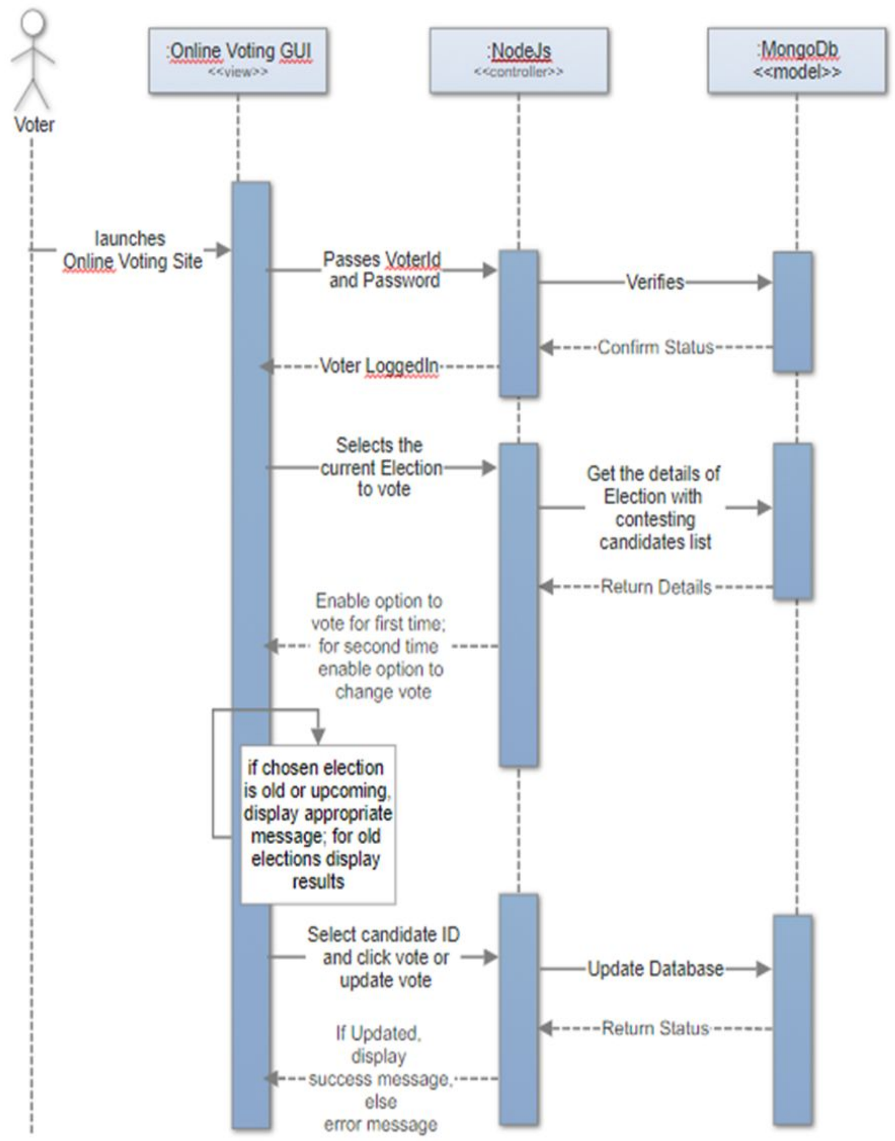
Sequence Diagram: Election Commission - Create Election


:Online Voting GUI
<<view>>
:NodeJs
<<controller>>
:MongoDb
<<model>>
Election Commission/Admin
launches Online Voting Site
Passes EC/AdminId and Password
Verifies
Confirm Status
EC/Admin LoggedIn
Selects the Create Election
Display create Election form with all necessary fields
Pass the details of Election with Start Date/End Date and type of election, etc
Process and Update Back end
Return Status
if created election already exists, display appropriate message to alert user.
If Created, display success message, else error message

Sequence Diagram: Candidate Nomination


Candidate/Party Head
:Online Voting GUI
<<view>>
:NodeJs
<<controller>>
:MongoDb
<<model>>
launches Online Voting Site
Passes candidateId and Password
Verifies
Confirm Status
Candidate LoggedIn
Selects the Election to contest
Get the details of Election with candidates nomination status
Return Details
Enable option to self nominate for candidate; for party head display option to nominate several candidates
if already nominated, display message, disable option
Enter candidate ID and click nominate
Update Back end
Return Status
If Updated, display success message, else error message

Sequence Diagram: Voter - Cast Vote


Voter
:Online Voting GUI
<<view>>
:NodeJs
<<controller>>
:MongoDb
<<model>>
launches Online Voting Site
Passes VoterId and Password
Verifies
Confirm Status
Voter LoggedIn
Selects the current Election to vote
Get the details of Election with contesting candidates list
Return Details
Enable option to vote for first time; for second time enable option to change vote
if chosen election is old or upcoming, display appropriate message; for old elections display results
Select candidate ID and click vote or update vote
Update Database
Return Status
If Updated, display success message, else error message

Iteration Plan 1

In the first iteration we have implemented the following:

1. **Create election**
2. **Vote**
3. **Nominate for Election**
4. **Update Vote**
5. **Update Voting Policy**
6. **Nominee Update Profile**

We were able to do all these 6 use cases in addition to these we were able to do these use cases as well:

1. **Register**
2. **Login**

Code Compilation

Please run "npm run start" in the compiled zip file or the full zip file to run the NodeJS server. The server will automatically connect to the database in mlab.com. Everything given in the screenshots below works as expected.

Database connection: mongo ds141175.mlab.com:41175/cs6905 -u student voting_system

If you run this in a command prompt and you have MongoDB installed, this will connect to the database at mlab.com and you will be able to see all the records there.

Test Accounts

All accounts password is "**test". All** the new nominations that are created have the password "**test"** as well. New accounts for nominations only gets created when that email account doesn't exists already. If it exists then that is nominated.

Election Official: official@voting.com

Candidates: candidate1@voting.com, candidate2@voting.com, candidate3@voting.com, candidate4@voting.com, candidate5@voting.com, candidate6@voting.com

Voters: voter1@voting.com, voter2@voting.com, voter3@voting.com, voter4@voting.com, voter5@voting.com, voter6@voting.com, voter7@voting.com, voter8@voting.com, voter9@voting.com

Database Schema

User Collection Schema

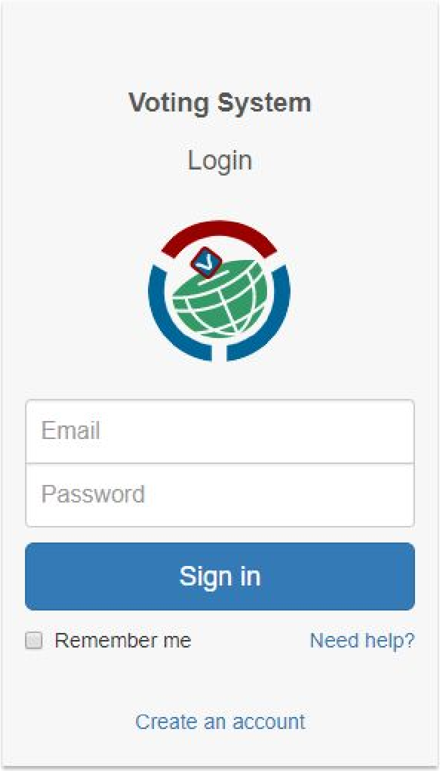
```
{
    "id": 1234567890,
    "email": "dummy@test.ca",
    "name": "Test User",
    "phone": "+1-322-555-5555",
    "address": "123 st street",
    "password": "$#2a#3asfdfdsfdsfdfsdfdsfdfs",
    "role": "electionOfficer",
    "securityQuestions": [
        {
            "question": "when were you born?",
            "answer": "Jan 1 1900"
        }
    ],
    "isVerified": true,
    "isEnabled": true,
    "incorrectLoginTries": 4,
    "lastActiveAt": "Sat Apr 08 2017 14:58:21 GMT-0400 (Eastern Daylight Time)",
    "createdAt": "Sat Apr 08 2017 14:58:21 GMT-0400 (Eastern Daylight Time)",
    "modifiedAt": "Sat Apr 08 2017 14:58:21 GMT-0400 (Eastern Daylight Time)"
}
```

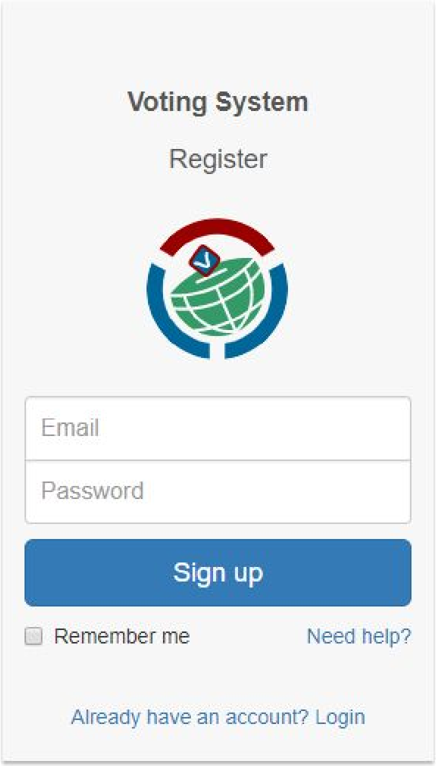
Election Collection Schema

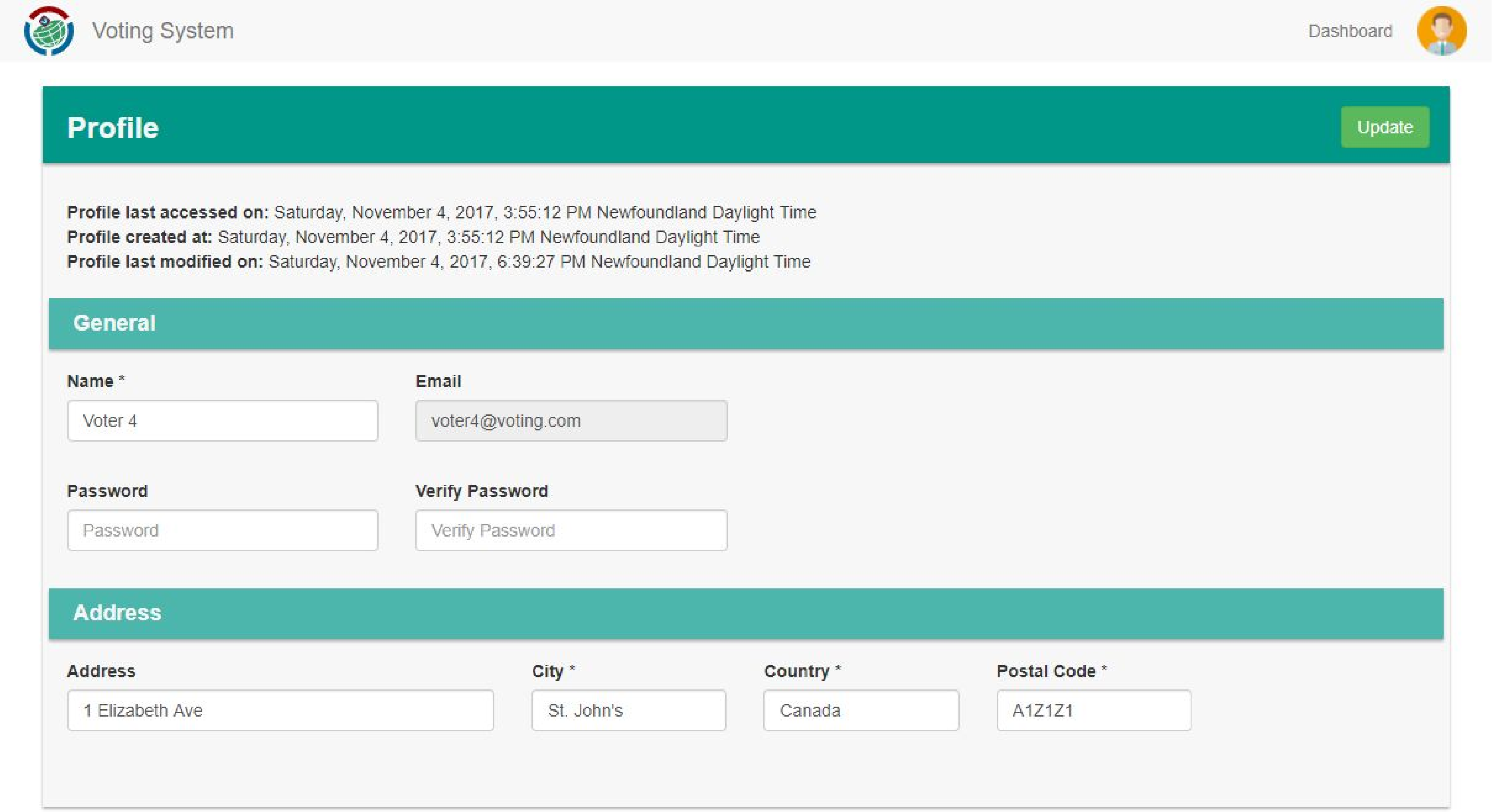
```
{
    "_id": 213454,
    "name": "Provincial Election - NL",
    "winningStrategy": "Winner Takes All",
    "dateFrom": "Thu Oct 13 2017 00:00:00 GMT-0230 (Newfoundland Daylight Time)",
    "dateTo": "Thu Oct 31 2018 00:00:00 GMT-0230 (Newfoundland Daylight Time)",
    "isDistrictElections": true,
    "numberOfDistricts": 5,
    "candidatesStrategy": "predefined",
    "usersStrategy": "registeredUsers",
    "userObjectConditions": "",
    "candidates": [
        {
            "candidateId": 234455,
            "numOfVotes": 123,
            "isApproved": false
        }
    ],
    "voters": [
        {
            "voterId": 234455,
            "hasVoted": false,
            "votedFor": 3456634
        }
    ]
}
```

Screenshots from Iteration 1

Login and Registration







Profile

Profile last accessed on: Saturday, November 4, 2017, 3:55:12 PM Newfoundland Daylight Time
Profile created at: Saturday, November 4, 2017, 3:55:12 PM Newfoundland Daylight Time
Profile last modified on: Saturday, November 4, 2017, 6:39:27 PM Newfoundland Daylight Time

General

Name * Voter 4

Email voter4@voting.com

Password Password

Verify Password Verify Password

Address

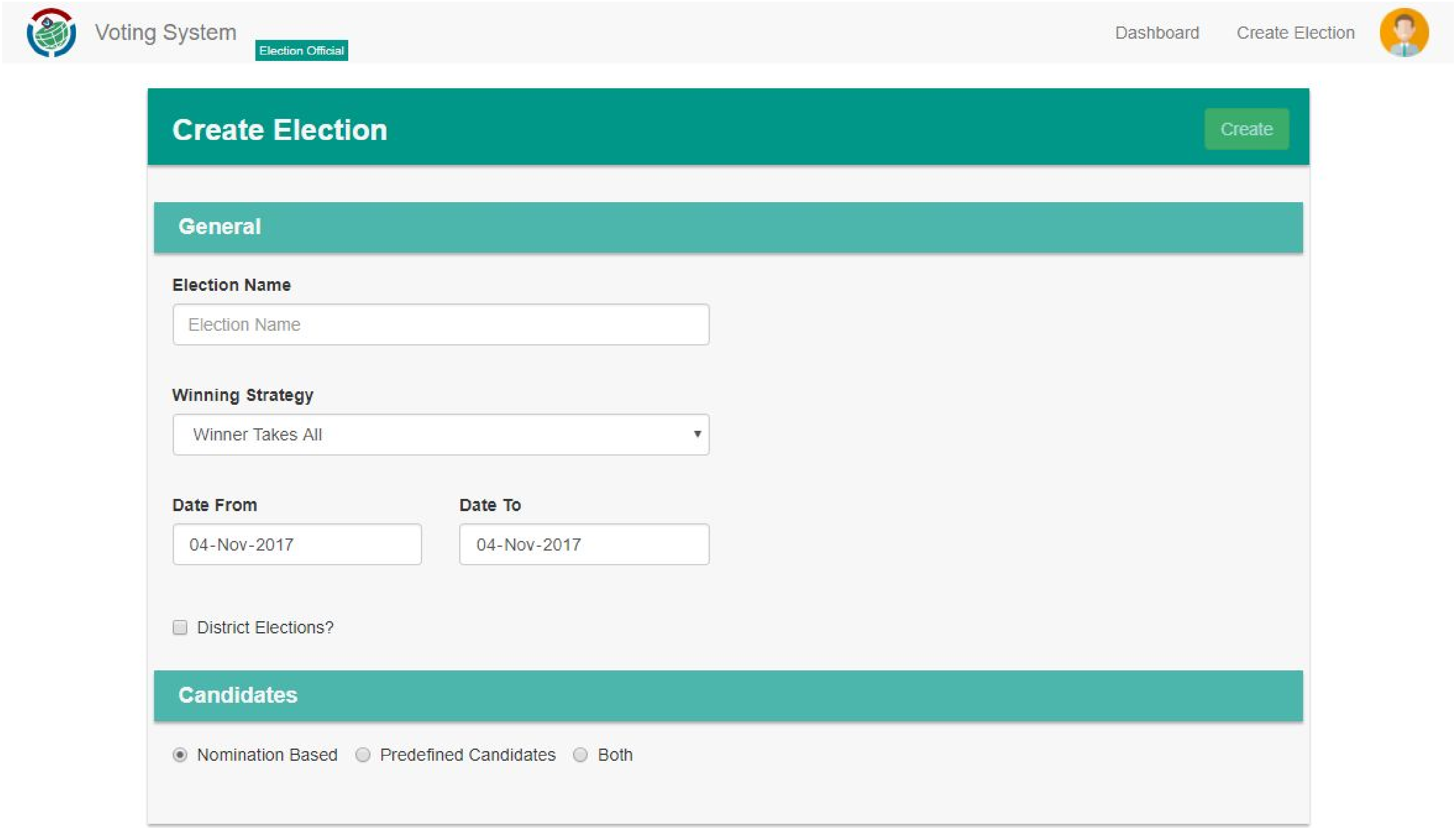
Address 1 Elizabeth Ave

City * St. John's

Country * Canada

Postal Code * A1Z1Z1

Election Official Perspective



Create Election

General

Election Name

Election Name

Winning Strategy

Winner Takes All ▾

Date From

04-Nov-2017

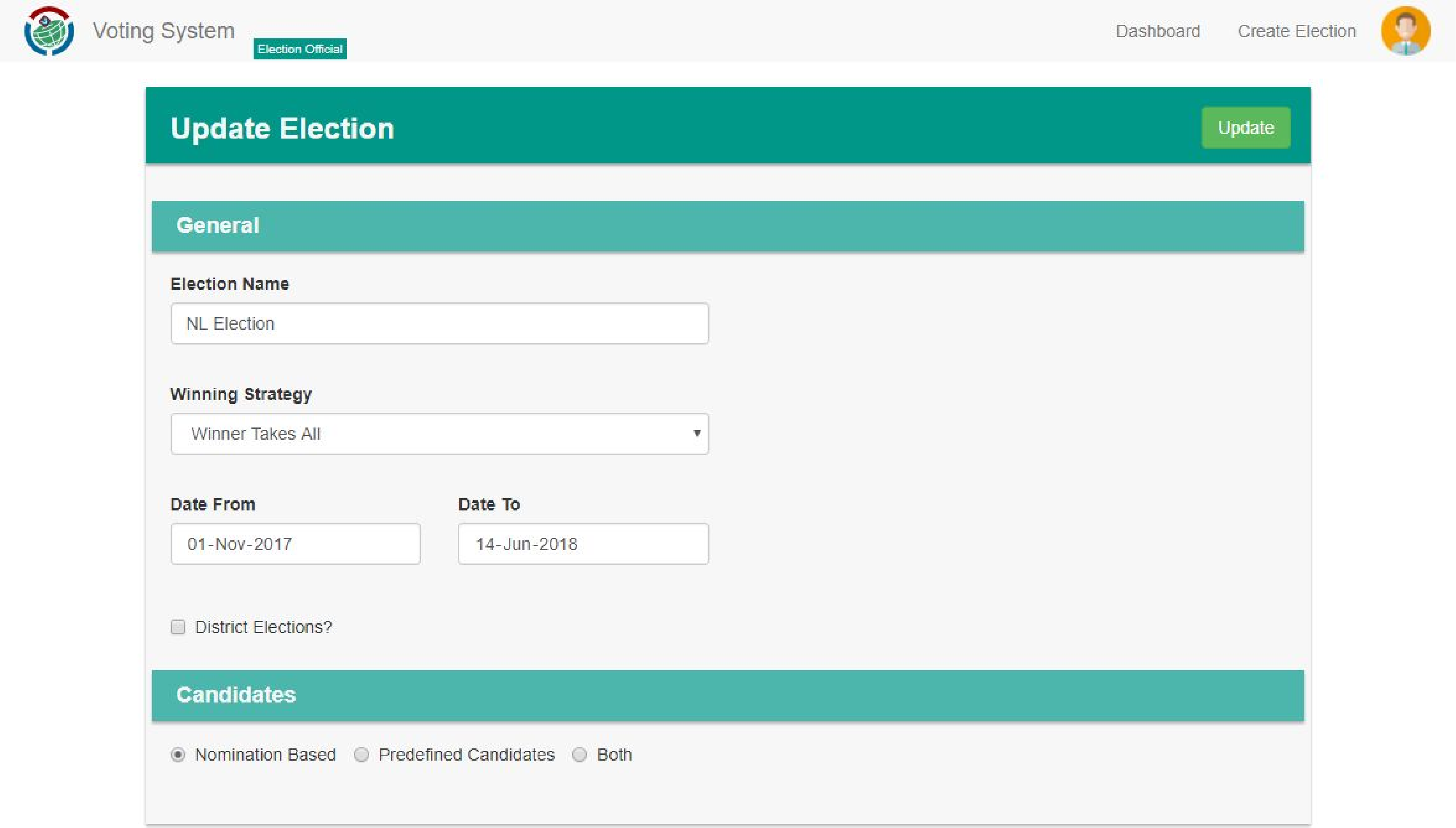
Date To

04-Nov-2017

□ District Elections?

Candidates

● Nomination Based ○ Predefined Candidates ○ Both



Update Election

Update

General

Election Name

NL Election

Winning Strategy

Winner Takes All ▾

Date From

01-Nov-2017

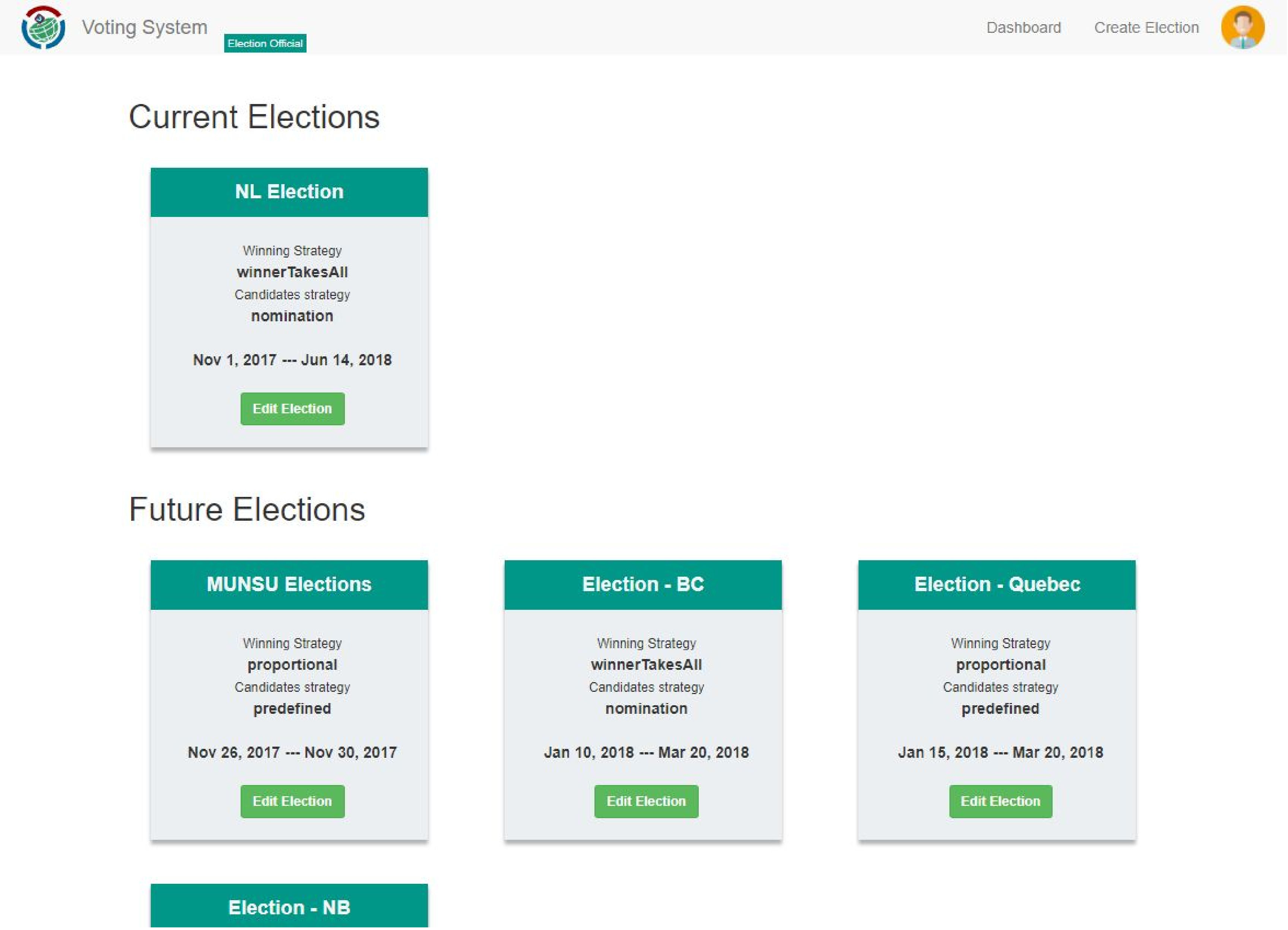
Date To

14-Jun-2018

□ District Elections?

Candidates

● Nomination Based ○ Predefined Candidates ○ Both



Current Elections

NL Election

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Nov 1, 2017 --- Jun 14, 2018

Future Elections

MUNSU Elections

Winning Strategy
proportional
Candidates strategy
predefined

Nov 26, 2017 --- Nov 30, 2017

Election - BC

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

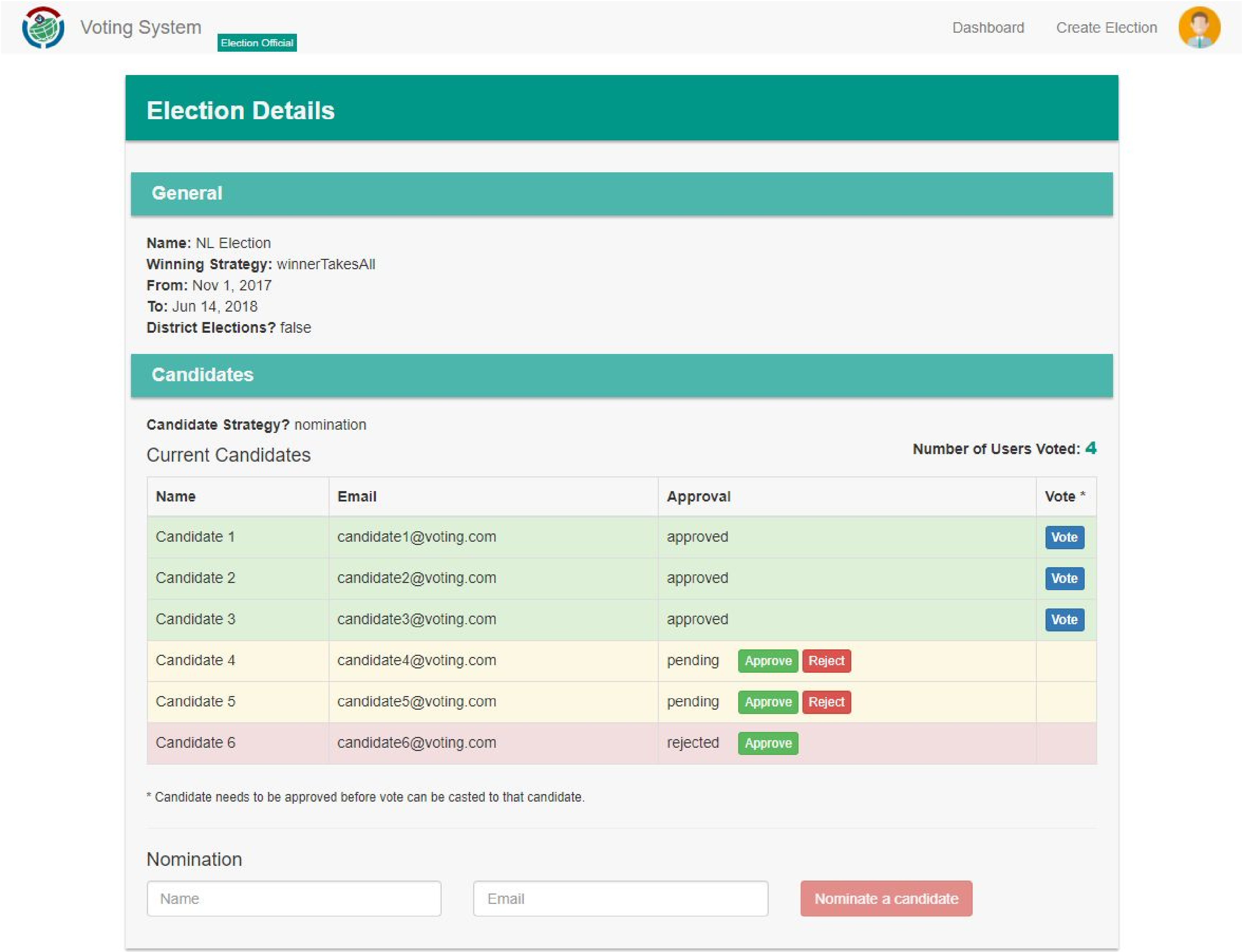
Jan 10, 2018 --- Mar 20, 2018

Election - Quebec

Winning Strategy
proportional
Candidates strategy
predefined

Jan 15, 2018 --- Mar 20, 2018

Election - NB



Election Details

General

Name: NL Election
Winning Strategy: winnerTakesAll
From: Nov 1, 2017
To: Jun 14, 2018
District Elections? false

Candidates

Candidate Strategy? nomination

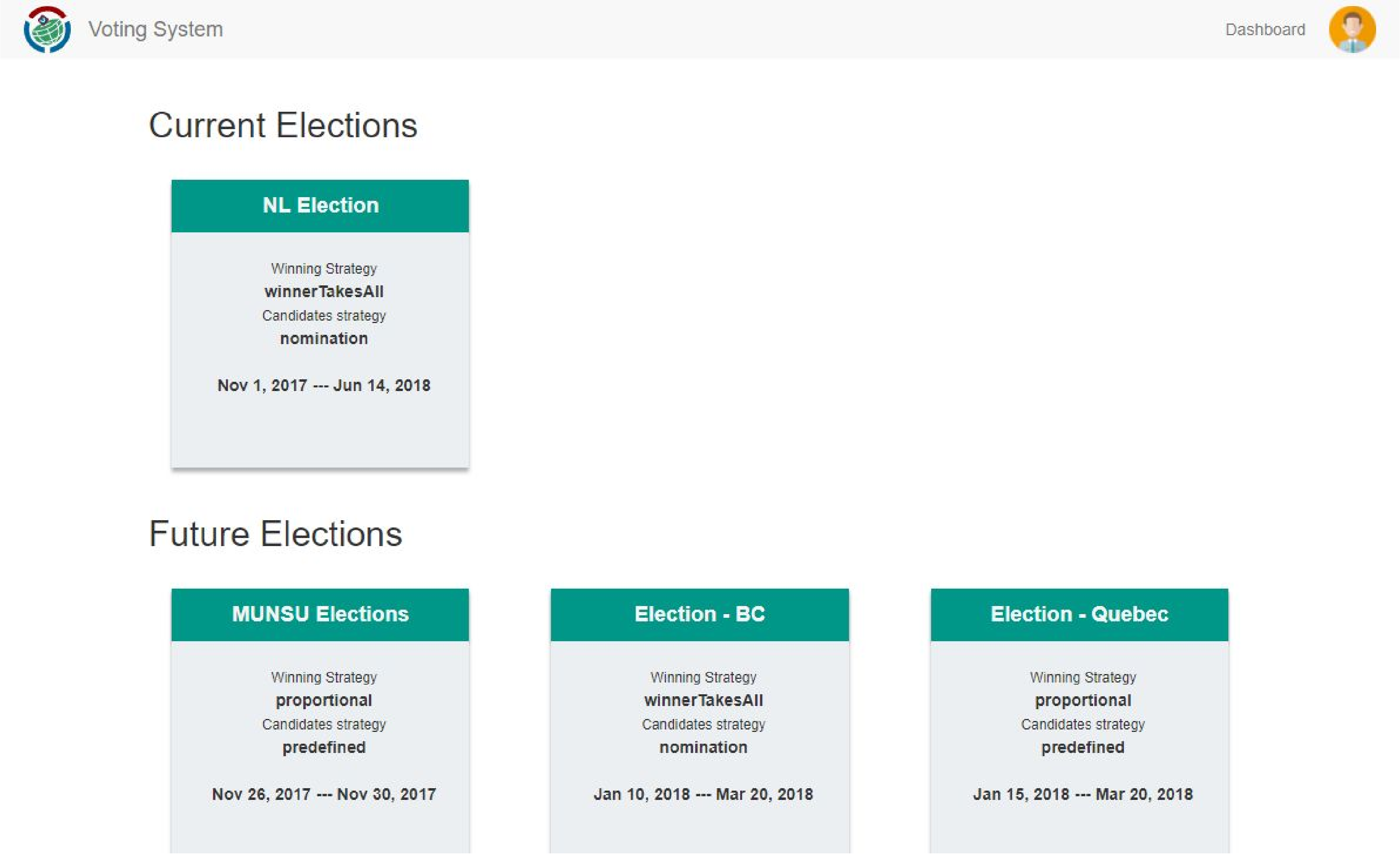
Current Candidates

Number of Users Voted: 4

Name	Email	Approval	Vote *
Candidate 1	candidate1@voting.com	approved	Vote
Candidate 2	candidate2@voting.com	approved	Vote
Candidate 3	candidate3@voting.com	approved	Vote
Candidate 4	candidate4@voting.com	pending Approve Reject	
Candidate 5	candidate5@voting.com	pending Approve Reject	
Candidate 6	candidate6@voting.com	rejected Approve	

* Candidate needs to be approved before vote can be casted to that candidate.

Nomination



Current Elections

NL Election

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Nov 1, 2017 --- Jun 14, 2018

Future Elections

MUNSU Elections

Winning Strategy
proportional
Candidates strategy
predefined

Nov 26, 2017 --- Nov 30, 2017

Election - BC

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

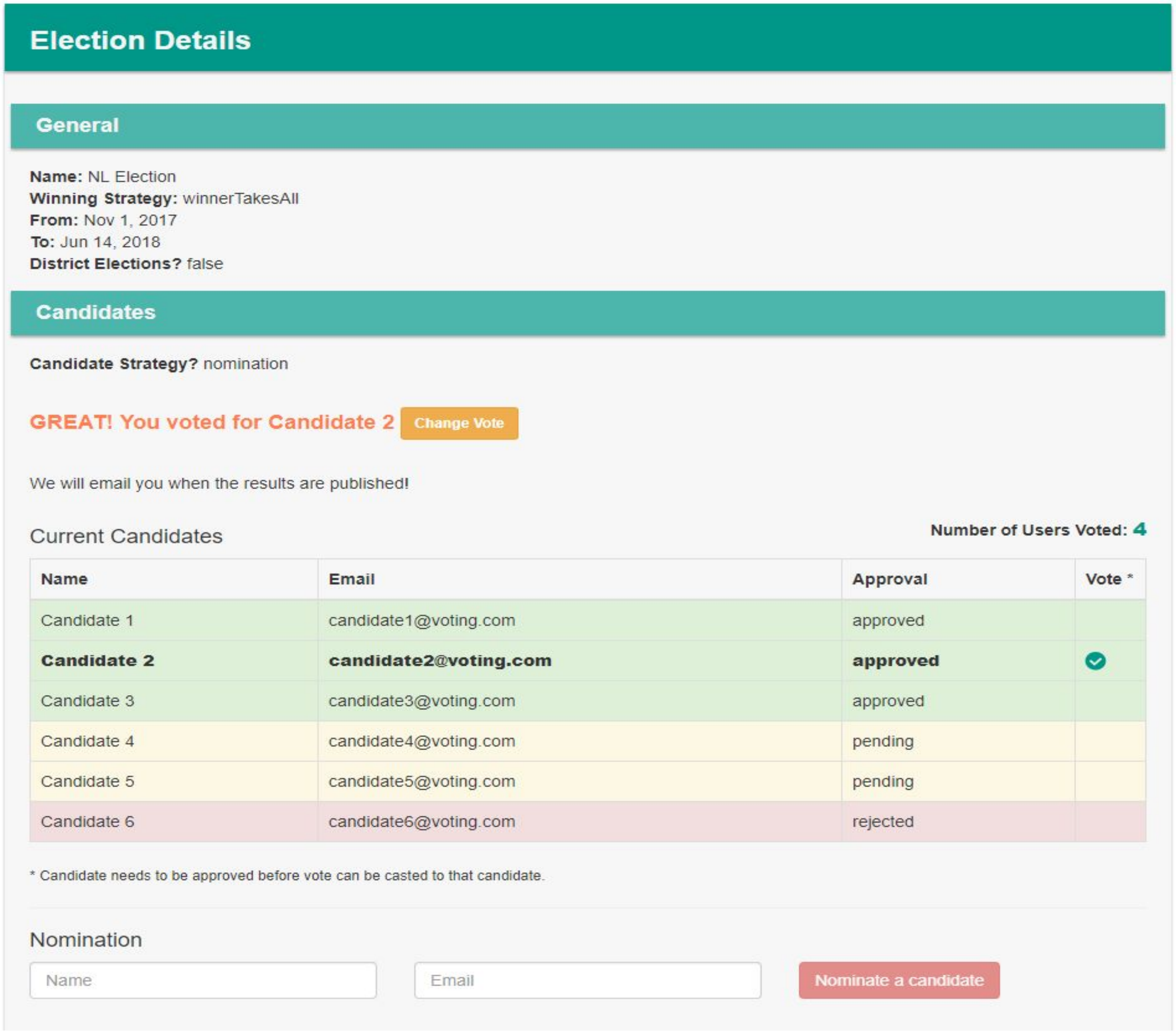
Jan 10, 2018 --- Mar 20, 2018

Election - Quebec

Winning Strategy
proportional
Candidates strategy
predefined

Jan 15, 2018 --- Mar 20, 2018

Voter Perspective



Election Details

General

Name: NL Election
Winning Strategy: winnerTakesAll
From: Nov 1, 2017
To: Jun 14, 2018
District Elections? false

Candidates

Candidate Strategy? nomination

GREAT! You voted for Candidate 2

We will email you when the results are published!

Current Candidates

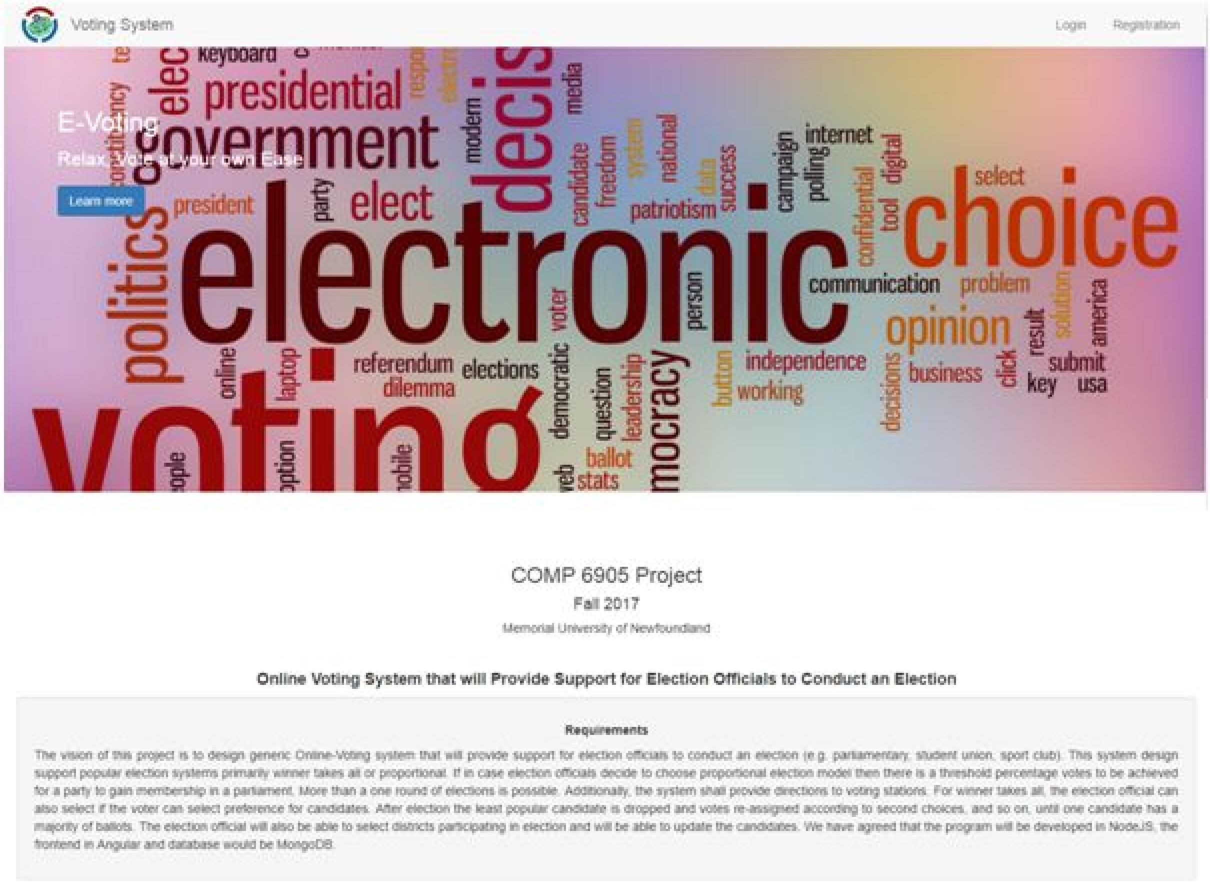
Number of Users Voted: 4

Name	Email	Approval	Vote *
Candidate 1	candidate1@voting.com	approved	
Candidate 2	**candidate2@voting.com**	**approved**	✓
Candidate 3	candidate3@voting.com	approved	
Candidate 4	candidate4@voting.com	pending	
Candidate 5	candidate5@voting.com	pending	
Candidate 6	candidate6@voting.com	rejected	

* Candidate needs to be approved before vote can be casted to that candidate.

Nomination

Some screenshots from the Final version



COMP 6905 Project

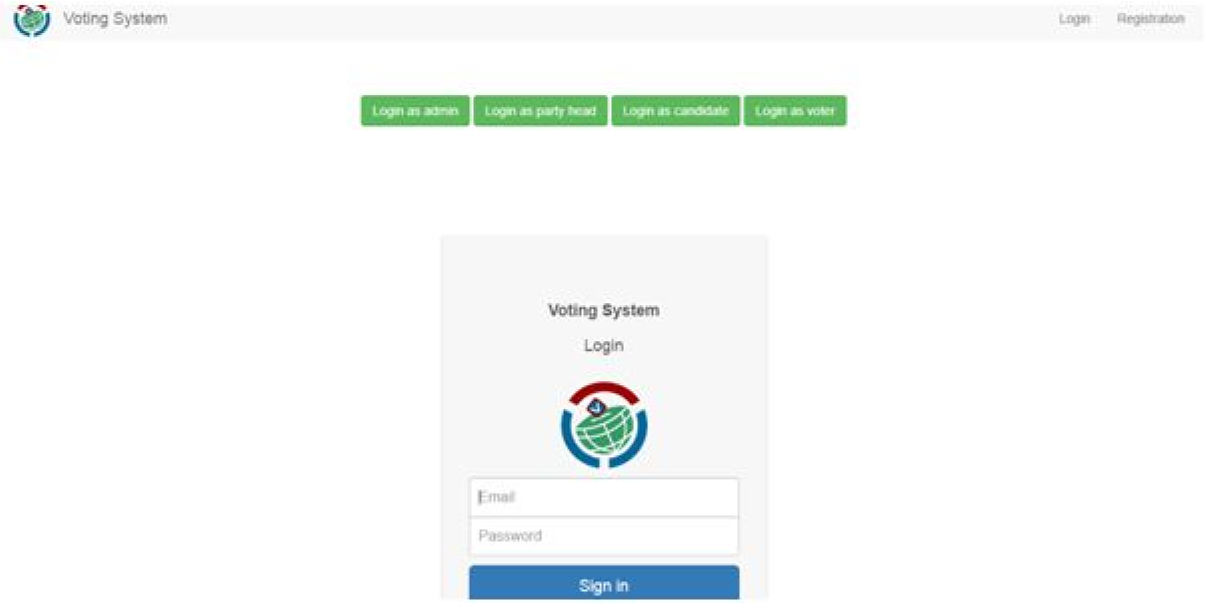
Fall 2017

Memorial University of Newfoundland

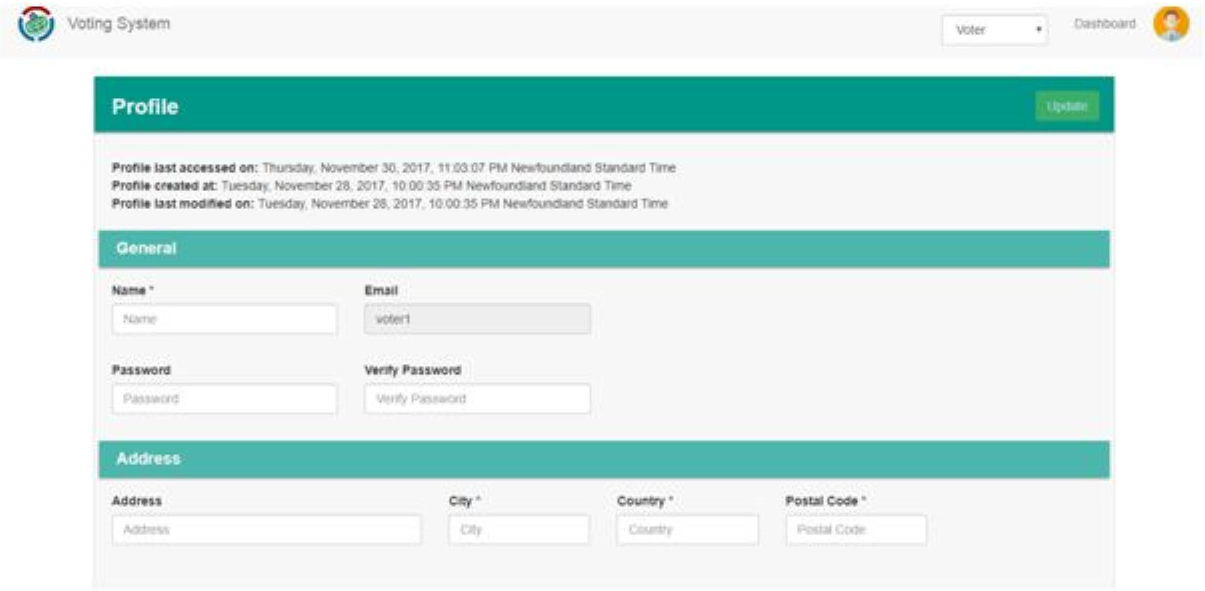
Online Voting System that will Provide Support for Election Officials to Conduct an Election

Requirements

The vision of this project is to design generic Online-Voting system that will provide support for election officials to conduct an election (e.g. parliamentary, student union, sport club). This system design support popular election systems primarily winner takes all or proportional. If in case election officals decide to choose proportional election model then there is a threshold percentage votes to be achieved for a party to gain membership in a parliament. More than a one round of elections is possible. Additionally, the system shall provide directions to voting stations. For winner takes all, the election official can also select if the voter can select preference for candidates. After election the least popular candidate is dropped and votes re-assigned according to second choices, and so on, until one candidate has a majority of ballots. The election official will also be able to select districts participating in election and will be able to update the candidates. We have agreed that the program will be developed in NodeJS, the frontend in Angular and database would be MongoDB.


Voting System
Login
Registration
Login as admin
Login as party head
Login as candidate
Login as voter
Voting System
Login
Email
Password
Sign in

Update the Profile



Profile

Profile last accessed on: Thursday, November 30, 2017, 11:03:07 PM Newfoundland Standard Time
Profile created at: Tuesday, November 28, 2017, 10:00:35 PM Newfoundland Standard Time
Profile last modified on: Tuesday, November 28, 2017, 10:00:35 PM Newfoundland Standard Time

General

Name *	Email
Name	voter1
Password	Verify Password
Password	Verify Password

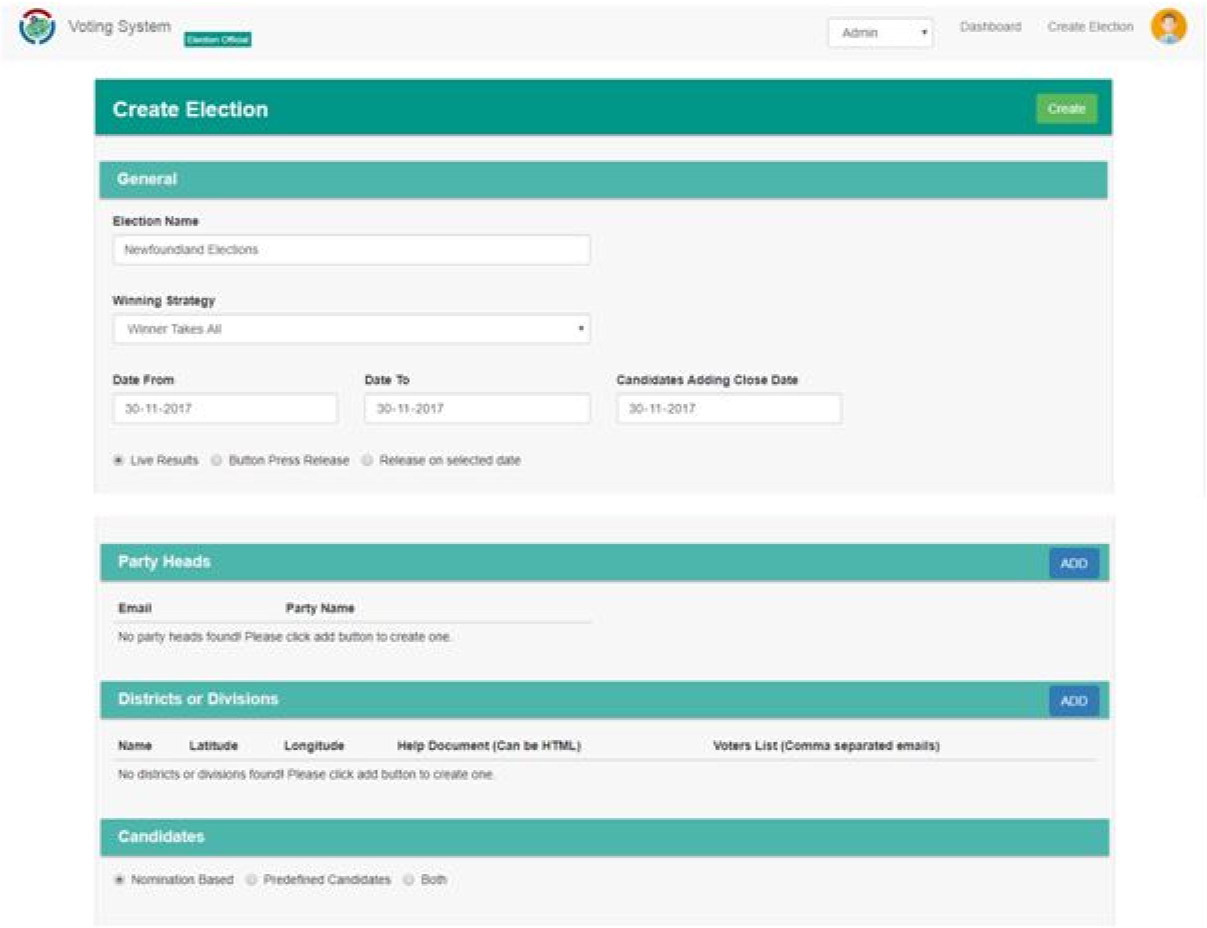
Address

Address	City *	Country *	Postal Code *
Address	City	Country	Postal Code

1. Admin Perspective

Create Election

- **Live Results**



Create Election

General

Election Name: Newfoundland Elections

Winning Strategy: Winner Takes All

Date From: 30-11-2017
Date To: 30-11-2017
Candidates Adding Close Date: 30-11-2017

● Live Results ○ Button Press Release ○ Release on selected date

Party Heads

Email	Party Name
No party heads found! Please click add button to create one.	

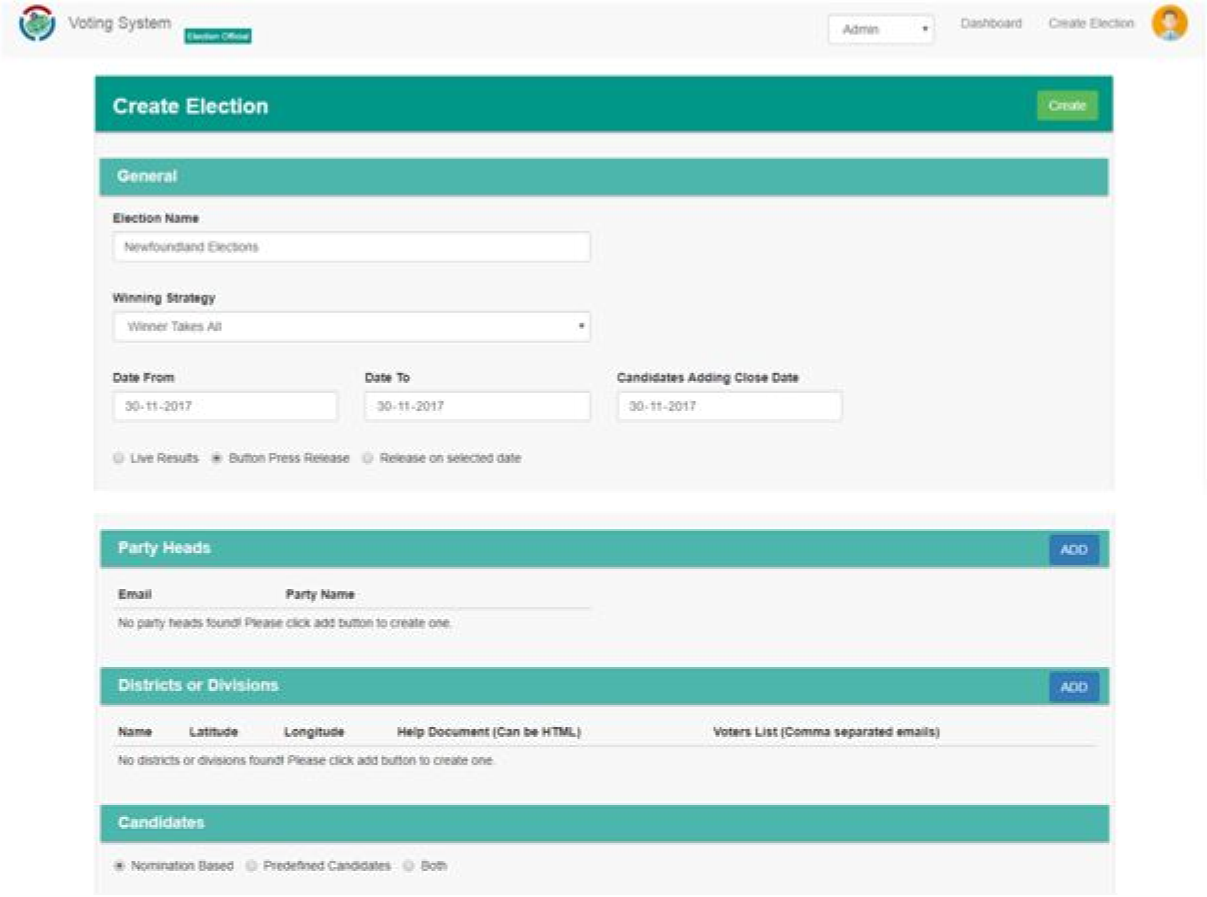
Districts or Divisions

Name	Latitude	Longitude	Help Document (Can be HTML)	Voters List (Comma separated emails)
No districts or divisions found! Please click add button to create one.				

Candidates

● Nomination Based ○ Predefined Candidates ○ Both

- **Button Press Release**



Create Election

General

Election Name: Newfoundland Elections

Winning Strategy: Winner Takes All

Date From	Date To	Candidates Adding Close Date
30-11-2017	30-11-2017	30-11-2017

○ Live Results ● Button Press Release ○ Release on selected date

Party Heads

Email	Party Name

No party heads found! Please click add button to create one.

Districts or Divisions

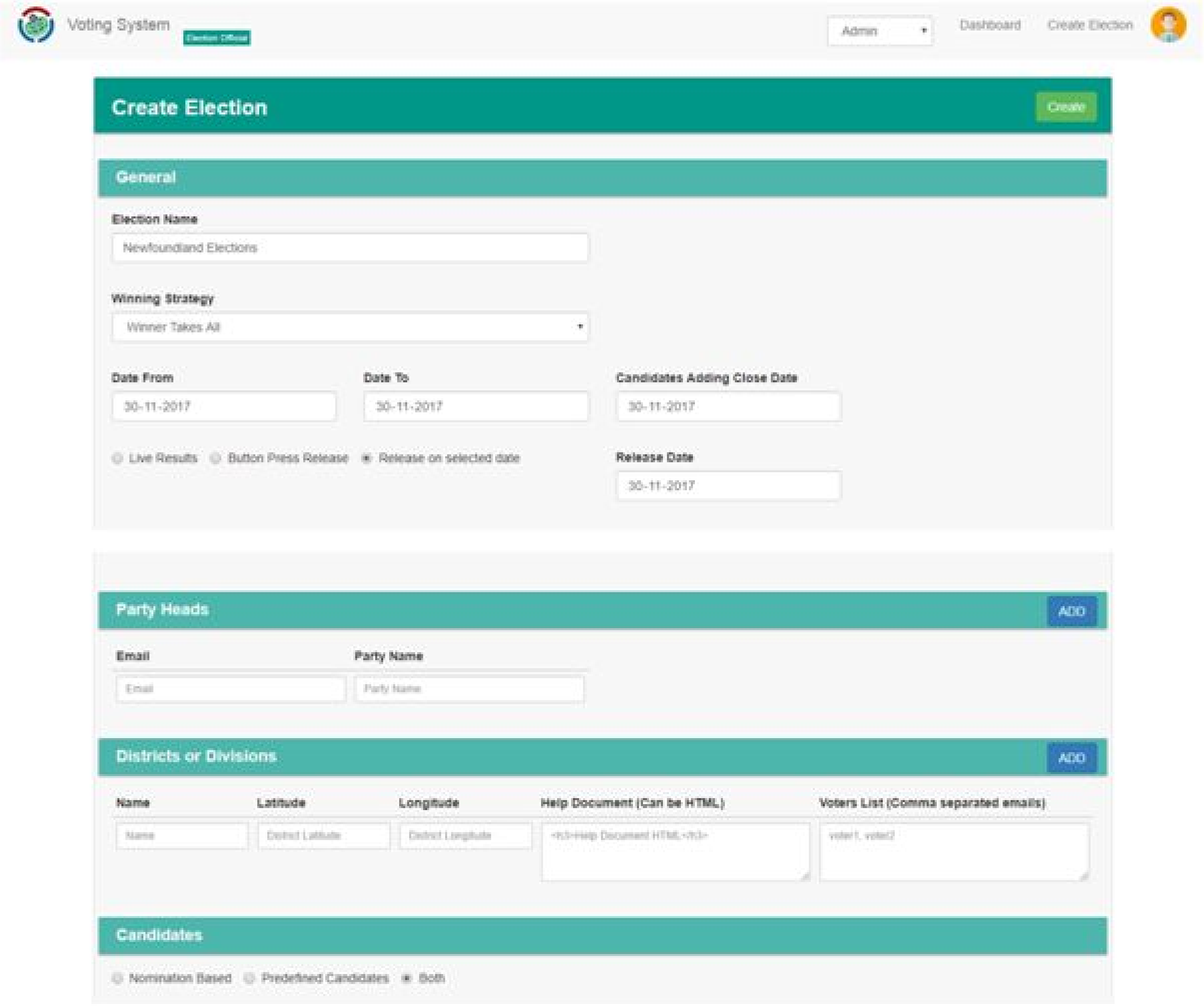
Name	Latitude	Longitude	Help Document (Can be HTML)	Voters List (Comma separated emails)

No districts or divisions found! Please click add button to create one.

Candidates

● Nomination Based ○ Predefined Candidates ○ Both

- **Release on Selected Date**



Create Election

General

Election Name: Newfoundland Elections

Winning Strategy: Winner Takes All

Date From: 30-11-2017

Date To: 30-11-2017

Candidates Adding Close Date: 30-11-2017

○ Live Results ○ Button Press Release ● Release on selected date

Release Date: 30-11-2017

Party Heads

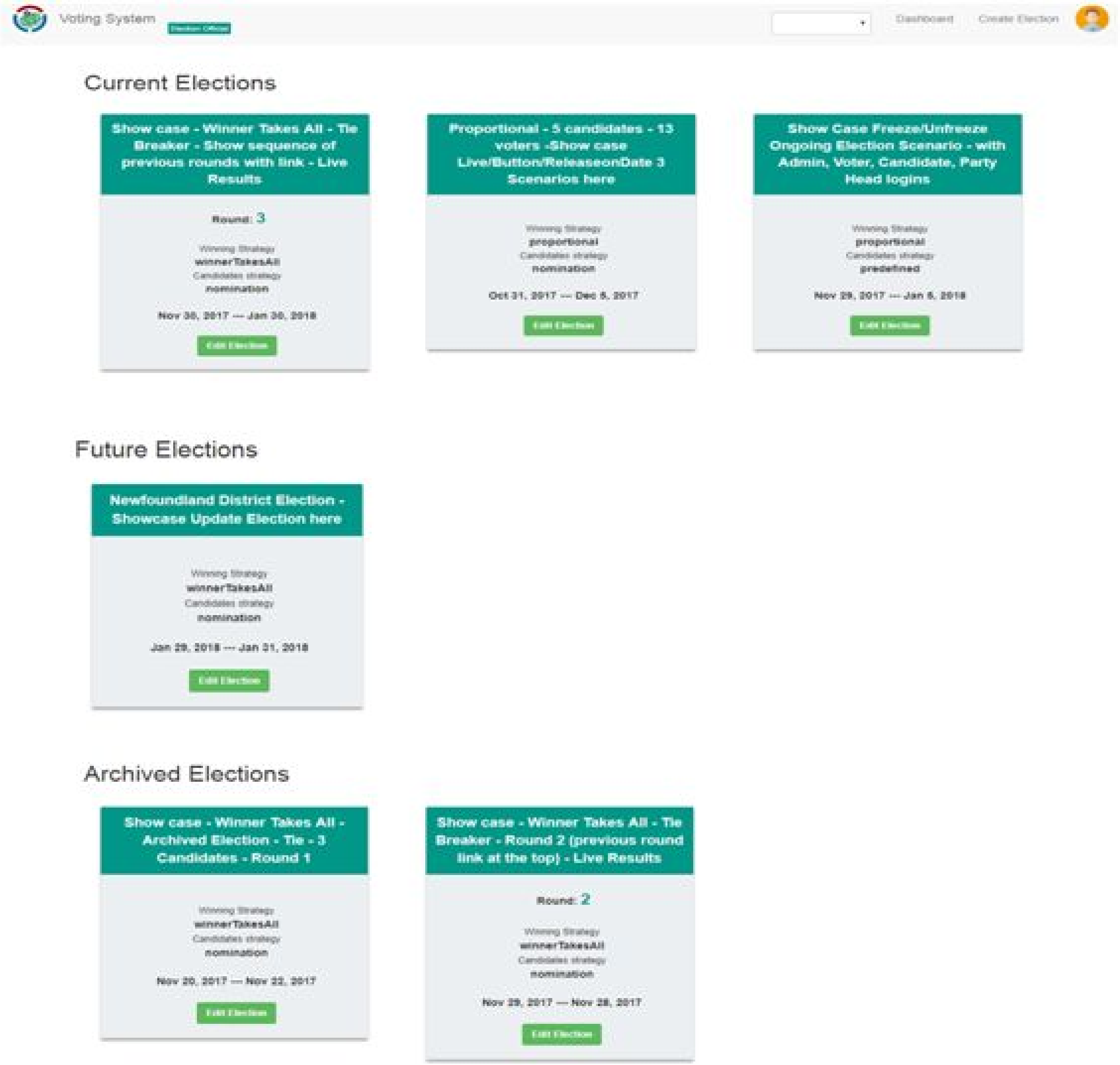
Email	Party Name
Email	Party Name

Districts or Divisions

Name	Latitude	Longitude	Help Document (Can be HTML)	Voters List (Comma separated emails)
Name	District Latitude	District Longitude	<h3>Help Document HTML</h3>	voter1, voter2

Candidates

○ Nomination Based ○ Predefined Candidates ● Both



Current Elections

Show case - Winner Takes All - Tie Breaker - Show sequence of previous rounds with link - Live Results

Round: 3

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Nov 30, 2017 — Jan 30, 2018

Edit Election

Proportional - 5 candidates - 13 voters -Show case Live/Button/ReleaseonDate 3 Scenarios here

Winning Strategy
proportional
Candidates strategy
nomination

Oct 31, 2017 — Dec 5, 2017

Edit Election

Show Case Freeze/Unfreeze Ongoing Election Scenario - with Admin, Voter, Candidate, Party Head logins

Winning Strategy
proportional
Candidates strategy
predefined

Nov 29, 2017 — Jan 5, 2018

Edit Election

Future Elections

Newfoundland District Election - Showcase Update Election here

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Jan 29, 2018 — Jan 31, 2018

Edit Election

Archived Elections

Show case - Winner Takes All - Archived Election - Tie - 3 Candidates - Round 1

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Nov 20, 2017 — Nov 22, 2017

Edit Election

Show case - Winner Takes All - Tie Breaker - Round 2 (previous round link at the top) - Live Results

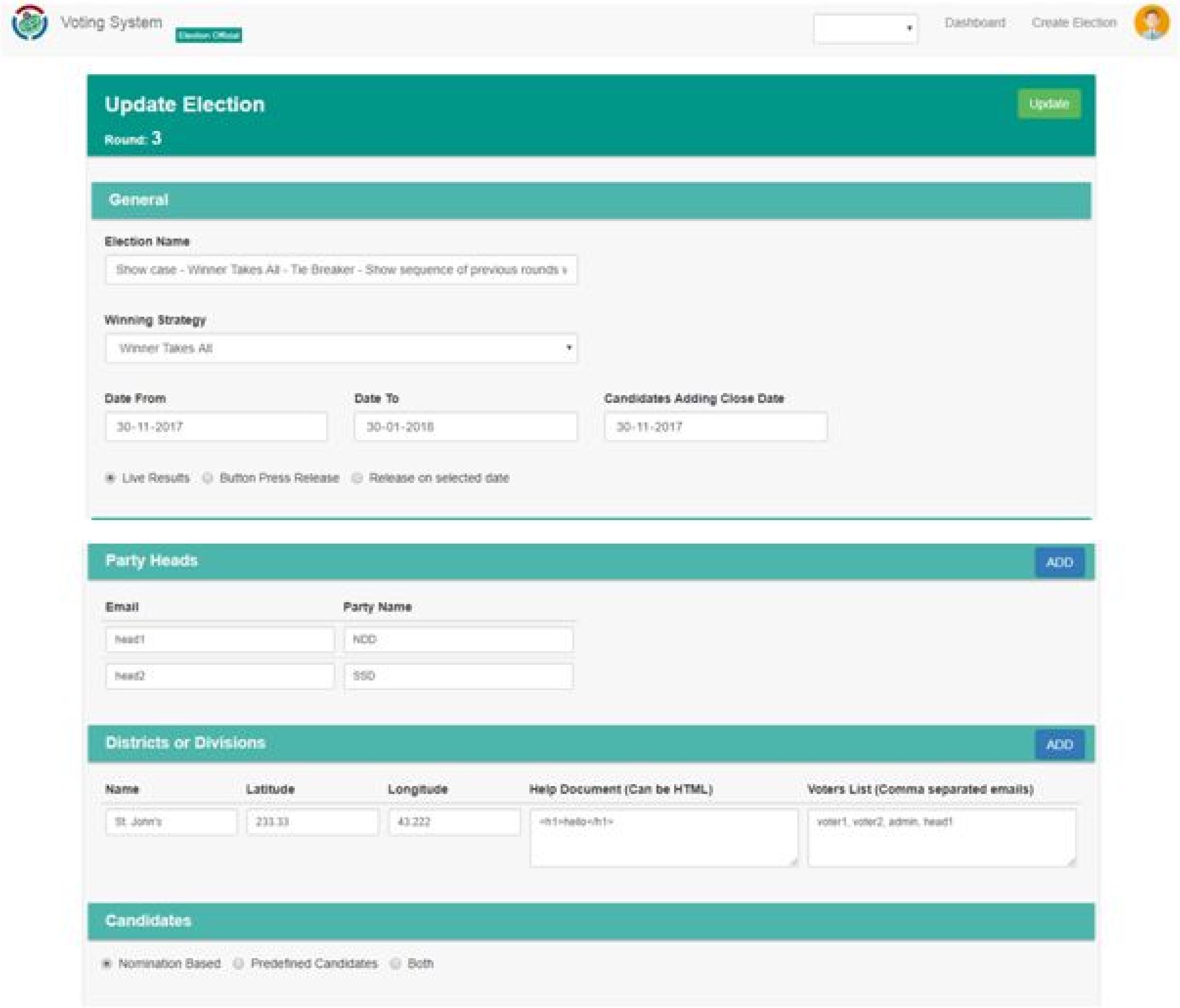
Round: 2

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Nov 29, 2017 — Nov 28, 2017

Edit Election

- **Update Election - Winner Takes All**



Update Election

Round: 3

General

Election Name: Show case - Winner Takes All - Tie Breaker - Show sequence of previous rounds v

Winning Strategy: Winner Takes All

Date From	Date To	Candidates Adding Close Date
30-11-2017	30-01-2018	30-11-2017

● Live Results ○ Button Press Release ○ Release on selected date

Party Heads

Email	Party Name
head1	NDD
head2	SSD

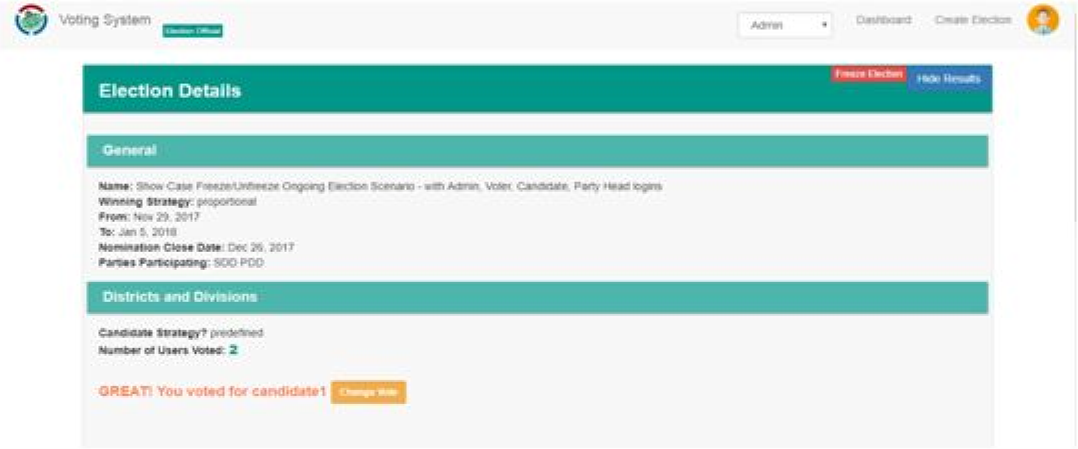
Districts or Divisions

Name	Latitude	Longitude	Help Document (Can be HTML)	Voters List (Comma separated emails)
St. John's	233.33	43.222	<h1>hello</h1>	voter1, voter2, admin, head1

Candidates

● Nomination Based ○ Predefined Candidates ○ Both

- **Freeze / Unfreeze Ongoing Elections**



Election Details

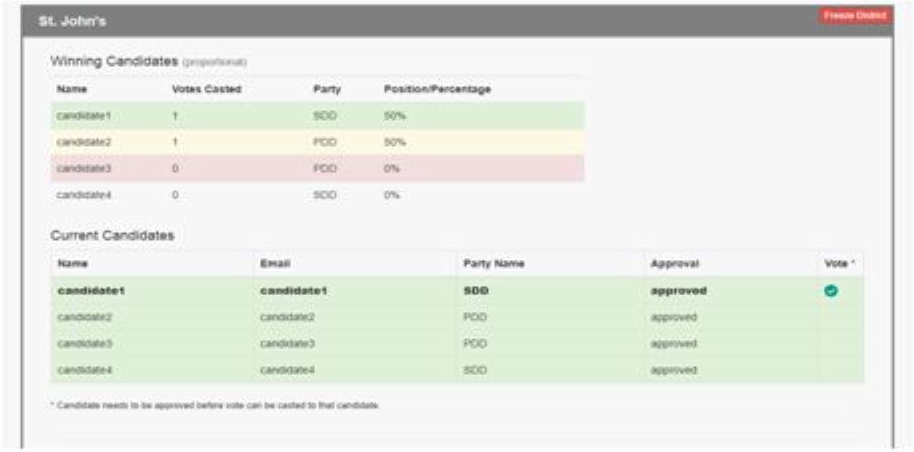
General

Name: Show Case Freeze/Unfreeze Ongoing Election Scenario - with Admin, Voter, Candidate, Party Head logins
Winning Strategy: proportional
From: Nov 29, 2017
To: Jan 5, 2018
Nomination Close Date: Dec 26, 2017
Parties Participating: SDD PDD

Districts and Divisions

Candidate Strategy? predefined
Number of Users Voted: 2

GREAT! You voted for candidate1



St. John's

Winning Candidates (proportional)

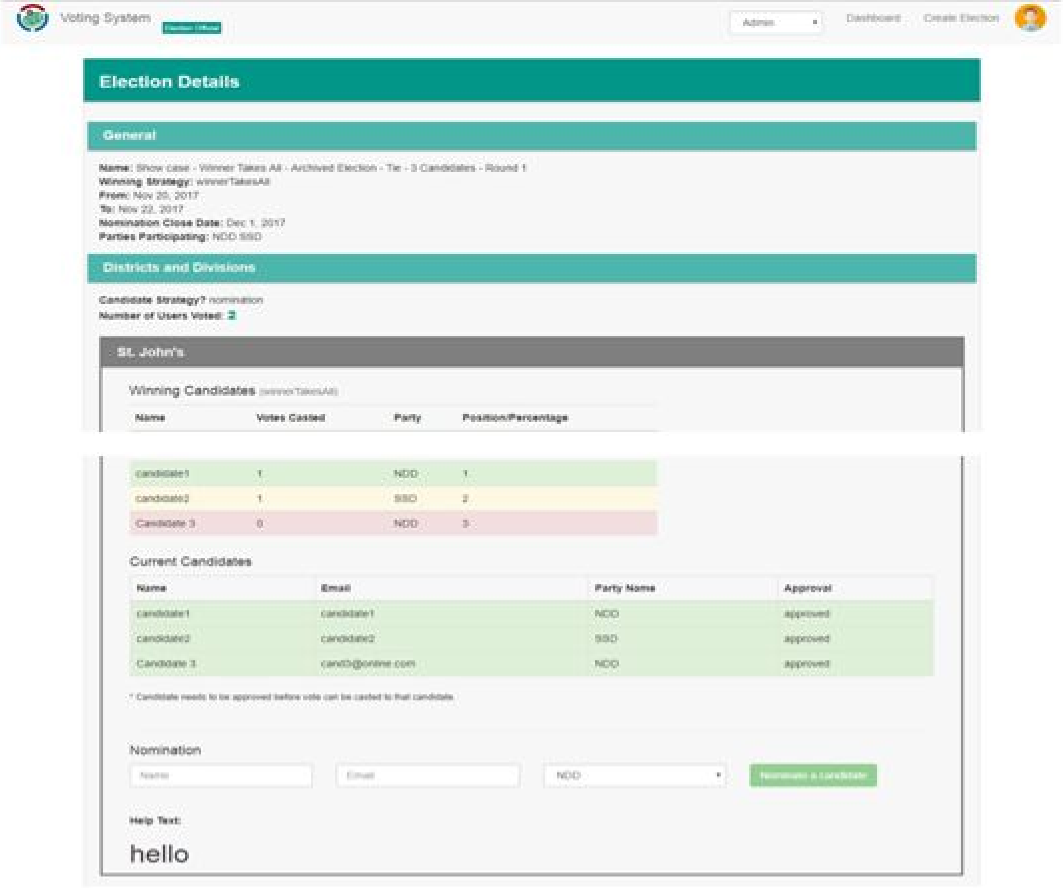
Name	Votes Casted	Party	Position/Percentage
candidate1	1	SDD	50%
candidate2	1	PDD	50%
candidate3	0	PDD	0%
candidate4	0	SDD	0%

Current Candidates

Name	Email	Party Name	Approval	Vote *
candidate1	**candidate1**	**SDD**	**approved**	✓
candidate2	candidate2	PDD	approved	
candidate3	candidate3	PDD	approved	
candidate4	candidate4	SDD	approved	

* Candidate needs to be approved before vote can be casted to that candidate.

- **Winner Takes All - Tie - Round 1 (3 candidates)**



Election Details

General

Name: Show case - Winner Takes All - Archived Election - Tie - 3 Candidates - Round 1
Winning Strategy: winnerTakesAll
From: Nov 20, 2017
To: Nov 22, 2017
Nomination Close Date: Dec 1, 2017
Parties Participating: NDD SSD

Districts and Divisions

Candidate Strategy? nomination
Number of Users Voted: 2

St. John's

Winning Candidates (winnerTakesAll)

Name	Votes Casted	Party	Position/Percentage
candidate1	1	NDD	1
candidate2	1	SSD	2
Candidate 3	0	NDD	3

Current Candidates

Name	Email	Party Name	Approval
candidate1	candidate1	NDD	approved
candidate2	candidate2	SSD	approved
Candidate 3	cand3@online.com	NDD	approved

* Candidate needs to be approved before vote can be casted to that candidate.

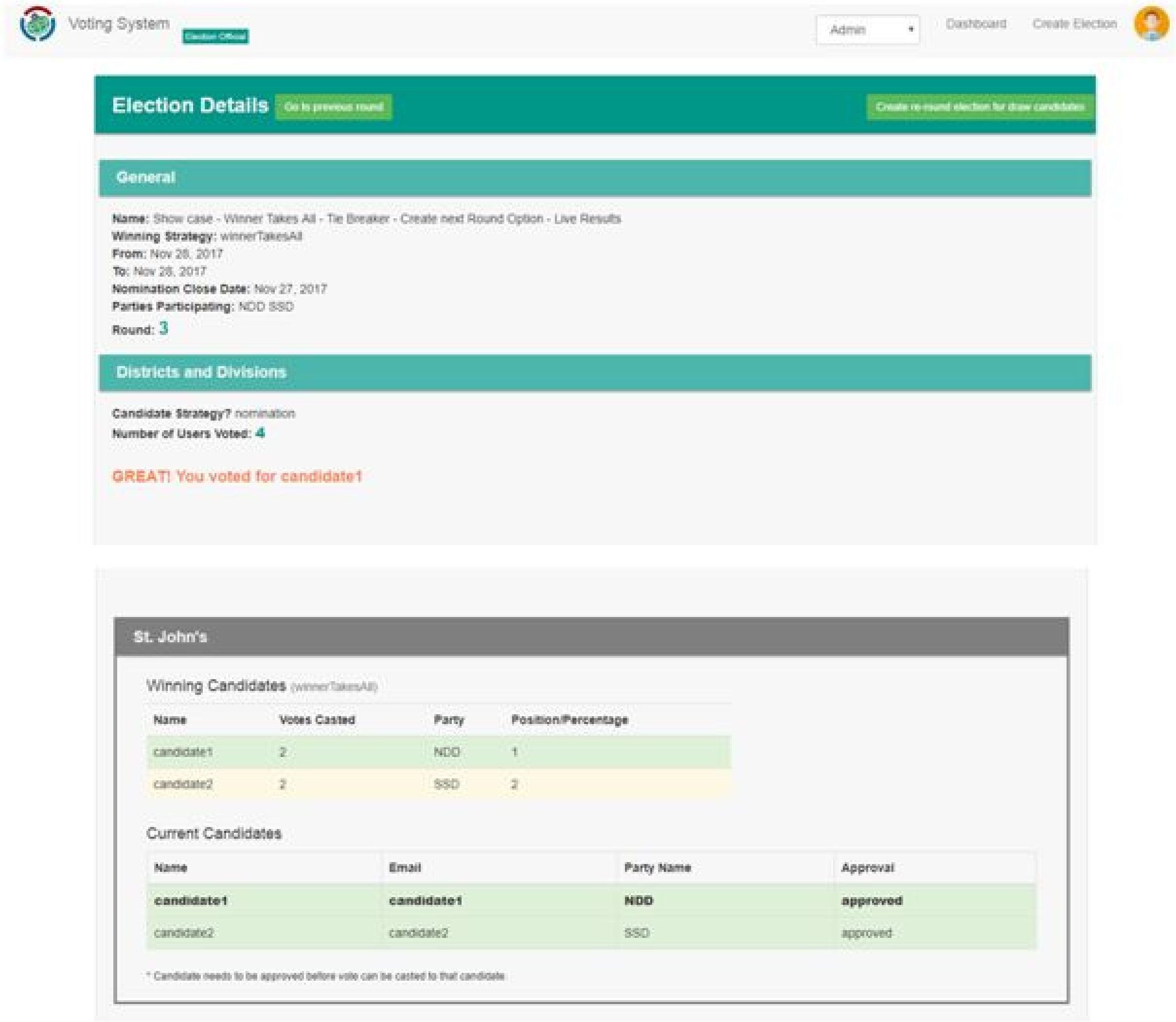
Nomination

Name | Email | NDD | Nominate a candidate

Help Text:

hello

- ## Winner Takes All - Tie Breaker - Next Round (Live Results)



Name: Show case - Winner Takes All - Tie Breaker - Create next Round Option - Live Results
Winning Strategy: winnerTakesAll
From: Nov 28, 2017
To: Nov 28, 2017
Nomination Close Date: Nov 27, 2017
Parties Participating: NDD SSD
Round: 3

Candidate Strategy? nomination
Number of Users Voted: 4

GREAT! You voted for candidate1

Winning Candidates (winnerTakesAll)

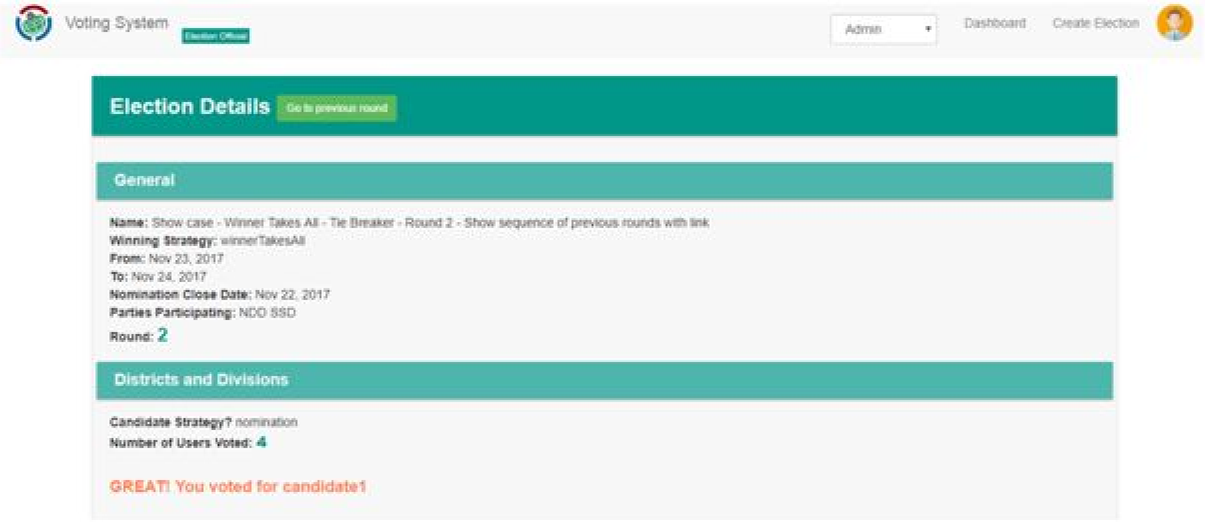
Name	Votes Casted	Party	Position/Percentage
candidate1	2	NDD	1
candidate2	2	SSD	2

Current Candidates

Name	Email	Party Name	Approval
candidate1	**candidate1**	**NDD**	**approved**
candidate2	candidate2	SSD	approved

* Candidate needs to be approved before vote can be casted to that candidate

- **Winner Takes All - Round 2: Show sequence of previous rounds with link**

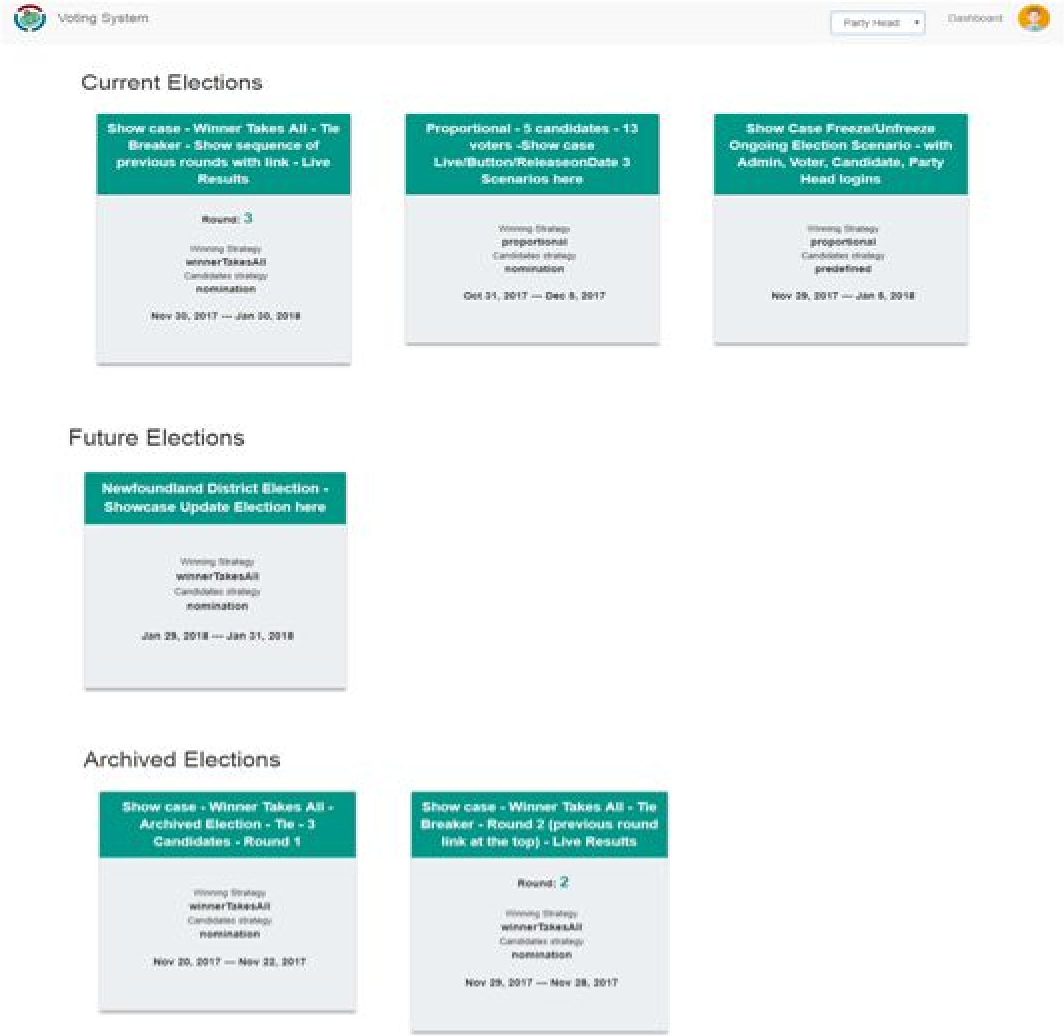


Name: Show case - Winner Takes All - Tie Breaker - Round 2 - Show sequence of previous rounds with link
Winning Strategy: winnerTakesAll
From: Nov 23, 2017
To: Nov 24, 2017
Nomination Close Date: Nov 22, 2017
Parties Participating: NDO SSD
Round: 2

Candidate Strategy? nomination
Number of Users Voted: 4

GREAT! You voted for candidate1

2. Party Head Perspective



Current Elections

Show case - Winner Takes All - Tie Breaker - Show sequence of previous rounds with link - Live Results
Round: 3
Winning Strategy
winnerTakesAll
Candidates strategy
nomination
Nov 30, 2017 — Jan 30, 2018

Proportional - 5 candidates - 13 voters -Show case Live/Button/ReleaseonDate 3 Scenarios here
Winning Strategy
proportional
Candidates strategy
nomination
Oct 31, 2017 — Dec 5, 2017

Show Case Freeze/Unfreeze Ongoing Election Scenario - with Admin, Voter, Candidate, Party Head logins
Winning Strategy
proportional
Candidates strategy
predefined
Nov 29, 2017 — Jan 8, 2018

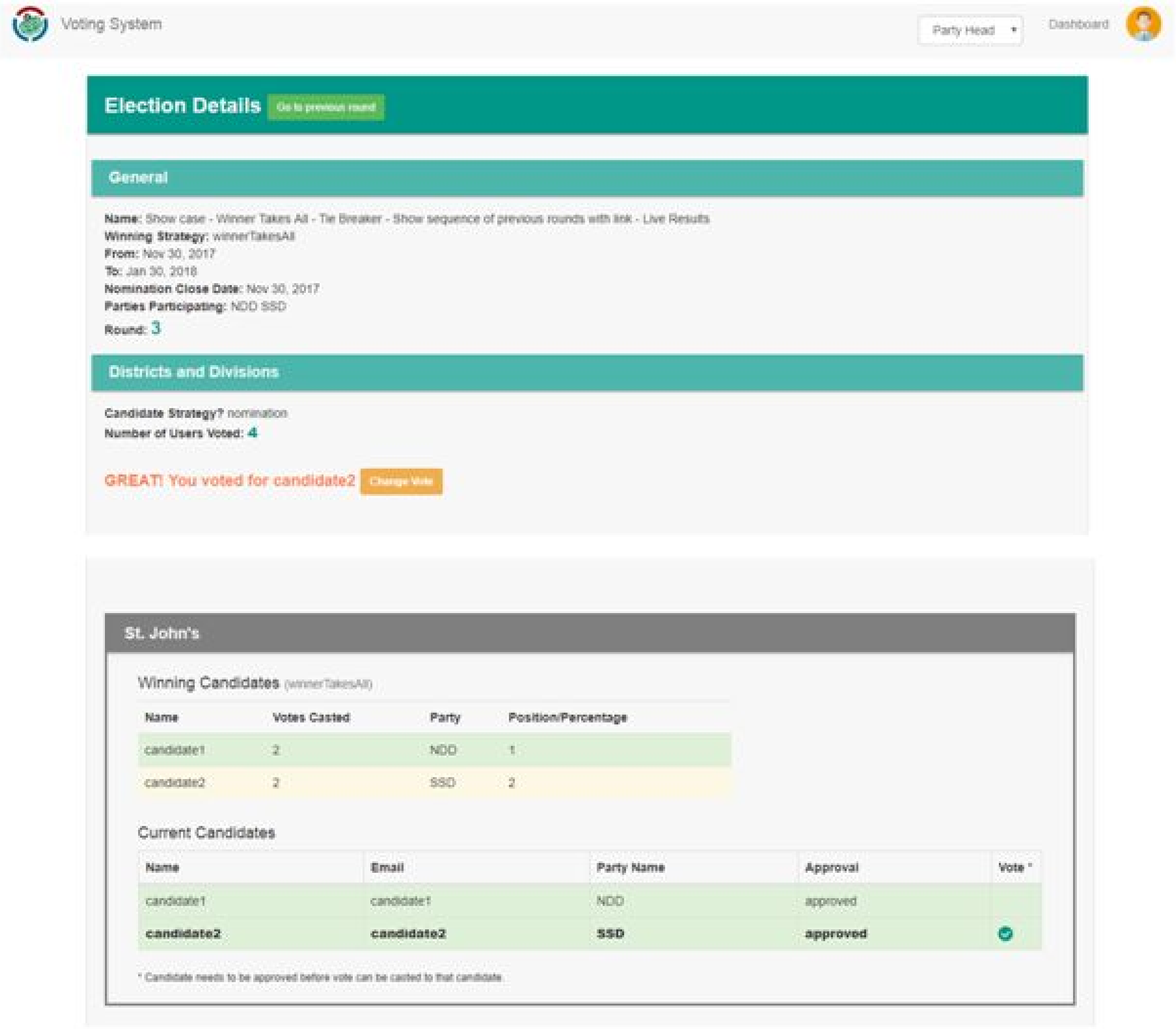
Future Elections

Newfoundland District Election - Showcase Update Election here
Winning Strategy
winnerTakesAll
Candidates strategy
nomination
Jan 29, 2018 — Jan 31, 2018

Archived Elections

Show case - Winner Takes All - Archived Election - Tie - 3 Candidates - Round 1
Winning Strategy
winnerTakesAll
Candidates strategy
nomination
Nov 20, 2017 — Nov 22, 2017

Show case - Winner Takes All - Tie Breaker - Round 2 (previous round link at the top) - Live Results
Round: 2
Winning Strategy
winnerTakesAll
Candidates strategy
nomination
Nov 29, 2017 — Nov 28, 2017



Election Details

General

Name: Show case - Winner Takes All - Tie Breaker - Show sequence of previous rounds with link - Live Results
Winning Strategy: winnerTakesAll
From: Nov 30, 2017
To: Jan 30, 2018
Nomination Close Date: Nov 30, 2017
Parties Participating: NDD SSD
Round: 3

Districts and Divisions

Candidate Strategy? nomination
Number of Users Voted: 4

GREAT! You voted for candidate2

St. John's

Winning Candidates (winnerTakesAll)

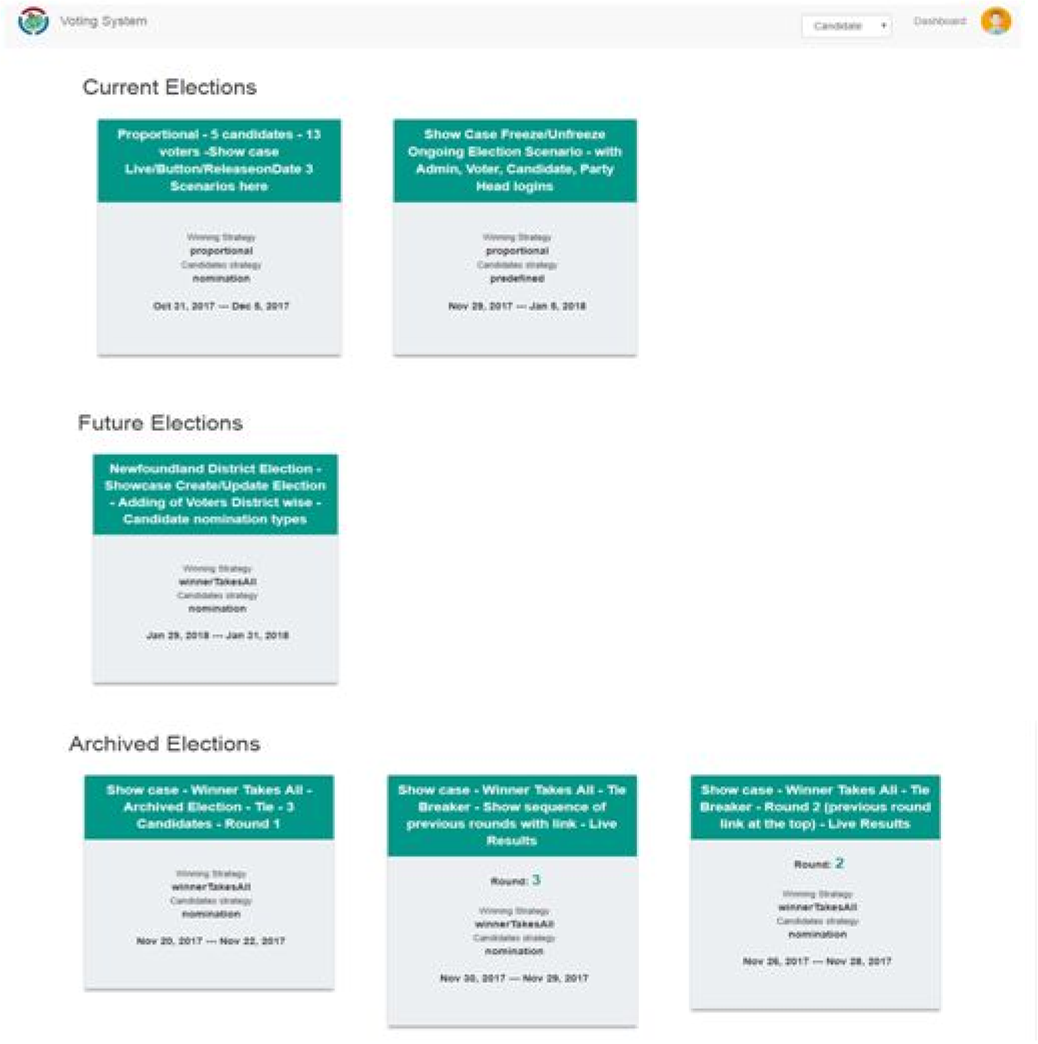
Name	Votes Casted	Party	Position/Percentage
candidate1	2	NDD	1
candidate2	2	SSD	2

Current Candidates

Name	Email	Party Name	Approval	Vote *
candidate1	candidate1	NDD	approved	
candidate2	**candidate2**	**SSD**	**approved**	✓

* Candidate needs to be approved before vote can be casted to that candidate.

3. Candidate Perspective



Current Elections

Proportional - 5 candidates - 13 voters -Show case Live/Button/ReleaseonDate 3 Scenarios here

Winning Strategy
proportional
Candidates strategy
nomination

Oct 21, 2017 — Dec 5, 2017

Show Case Freeze/Unfreeze Ongoing Election Scenario - with Admin, Voter, Candidate, Party Head logins

Winning Strategy
proportional
Candidates strategy
predefined

Nov 29, 2017 — Jan 5, 2018

Future Elections

Newfoundland District Election - Showcase Create/Update Election - Adding of Voters District wise - Candidate nomination types

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Jan 29, 2018 — Jan 31, 2018

Archived Elections

Show case - Winner Takes All - Archived Election - Tie - 3 Candidates - Round 1

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Nov 20, 2017 — Nov 22, 2017

Show case - Winner Takes All - Tie Breaker - Show sequence of previous rounds with link - Live Results

Round: 3

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

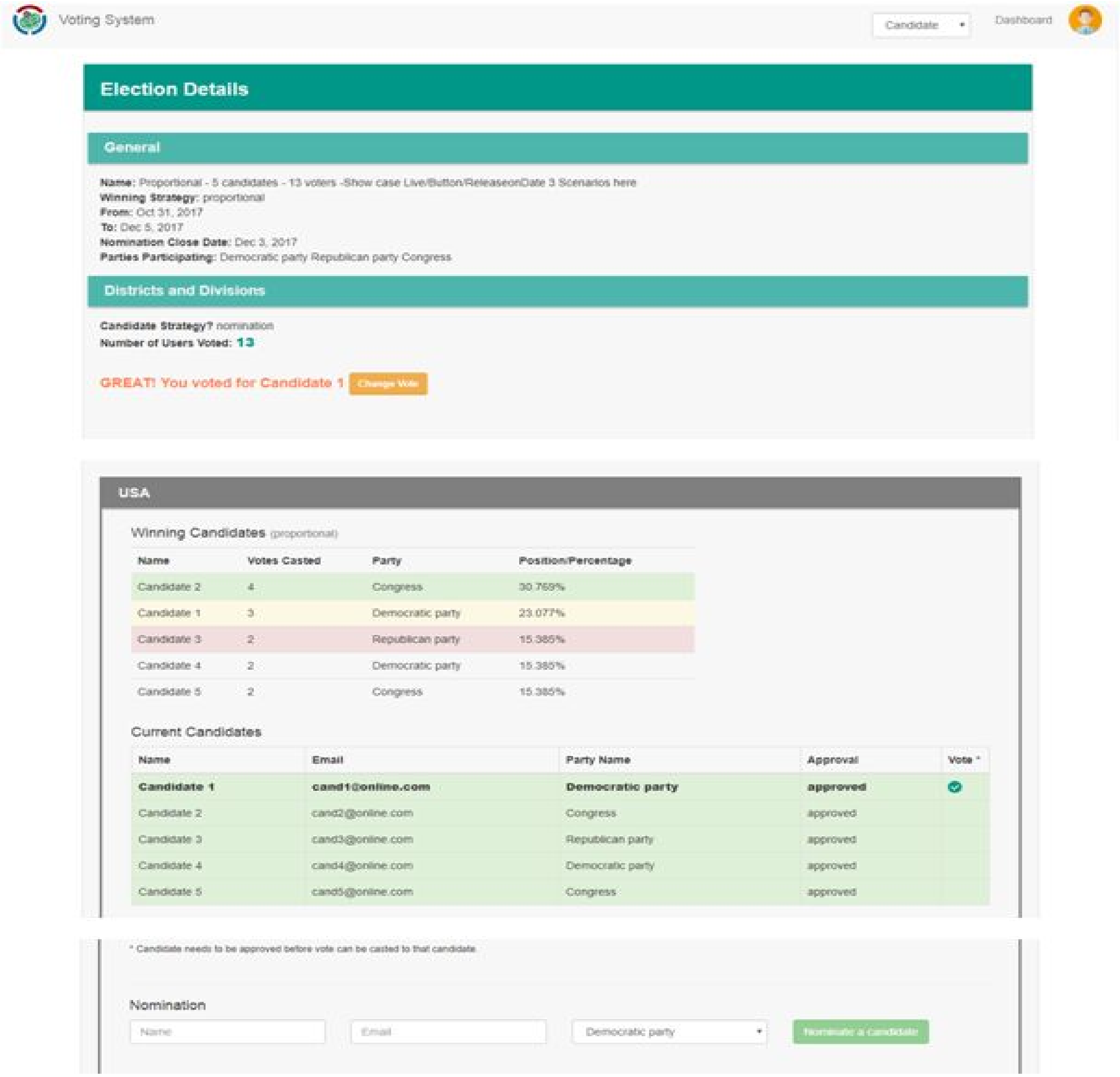
Nov 30, 2017 — Nov 29, 2017

Show case - Winner Takes All - Tie Breaker - Round 2 (previous round link at the top) - Live Results

Round: 2

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Nov 26, 2017 — Nov 28, 2017



Election Details

General

Name: Proportional - 5 candidates - 13 voters -Show case Live/Button/ReleaseonDate 3 Scenarios here
Winning Strategy: proportional
From: Oct 31, 2017
To: Dec 5, 2017
Nomination Close Date: Dec 3, 2017
Parties Participating: Democratic party Republican party Congress

Districts and Divisions

Candidate Strategy? nomination
Number of Users Voted: 13

GREAT! You voted for Candidate 1

USA

Winning Candidates (proportional)

Name	Votes Casted	Party	Position/Percentage
Candidate 2	4	Congress	30.769%
Candidate 1	3	Democratic party	23.077%
Candidate 3	2	Republican party	15.385%
Candidate 4	2	Democratic party	15.385%
Candidate 5	2	Congress	15.385%

Current Candidates

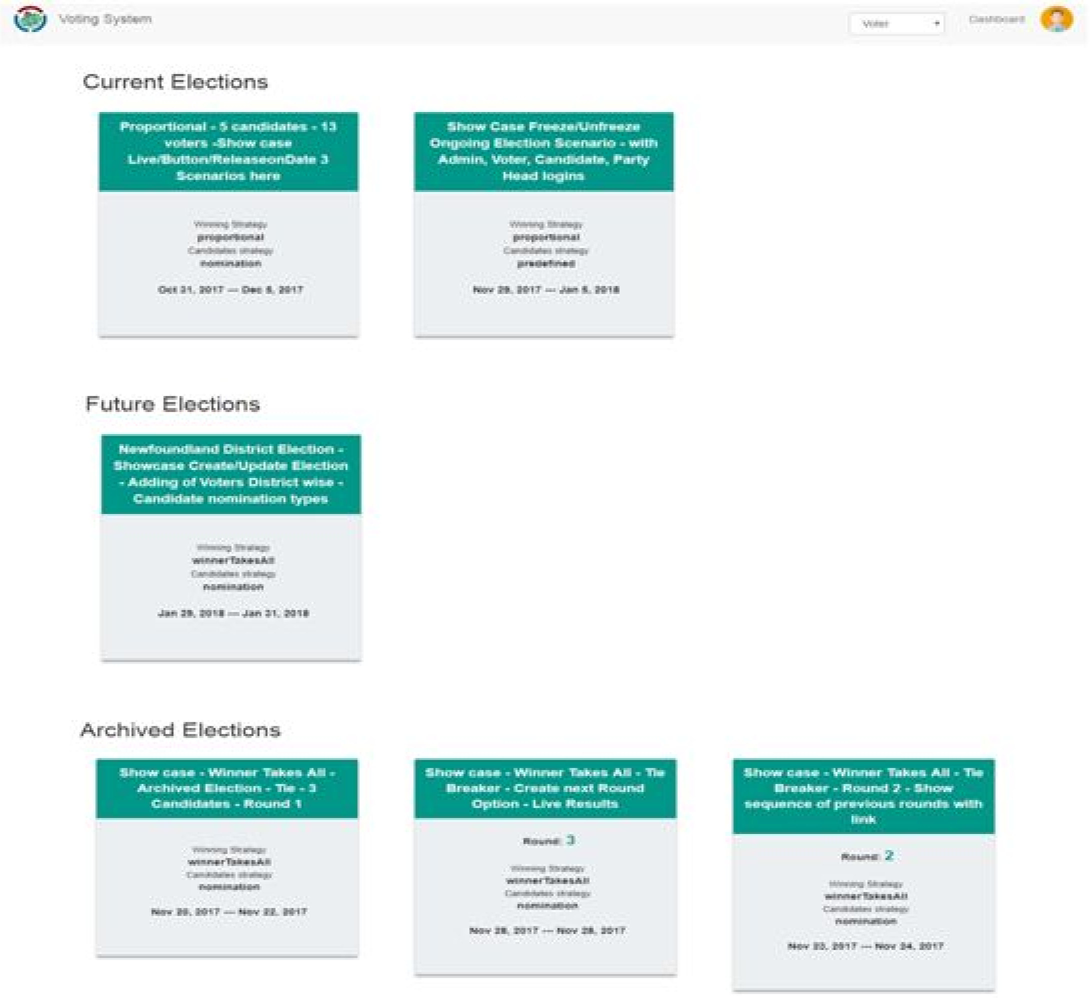
Name	Email	Party Name	Approval	Vote *
Candidate 1	**cand1@online.com**	**Democratic party**	**approved**	✓
Candidate 2	cand2@online.com	Congress	approved	
Candidate 3	cand3@online.com	Republican party	approved	
Candidate 4	cand4@online.com	Democratic party	approved	
Candidate 5	cand5@online.com	Congress	approved	

* Candidate needs to be approved before vote can be casted to that candidate.

Nomination

Name: ____ Email: ____ Democratic party ▾

4. Voter Perspective



Current Elections

Proportional - 5 candidates - 13 voters -Show case Live/Button/ReleaseonDate 3 Scenarios here

Winning Strategy
proportional
Candidates strategy
nomination

Oct 31, 2017 — Dec 5, 2017

Show Case Freeze/Unfreeze Ongoing Election Scenario - with Admin, Voter, Candidate, Party Head logins

Winning Strategy
proportional
Candidates strategy
predefined

Nov 29, 2017 — Jan 5, 2018

Future Elections

Newfoundland District Election - Showcase Create/Update Election - Adding of Voters District wise - Candidate nomination types

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Jan 29, 2018 — Jan 31, 2018

Archived Elections

Show case - Winner Takes All - Archived Election - Tie - 3 Candidates - Round 1

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Nov 20, 2017 — Nov 22, 2017

Show case - Winner Takes All - Tie Breaker - Create next Round Option - Live Results

Round: 3

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

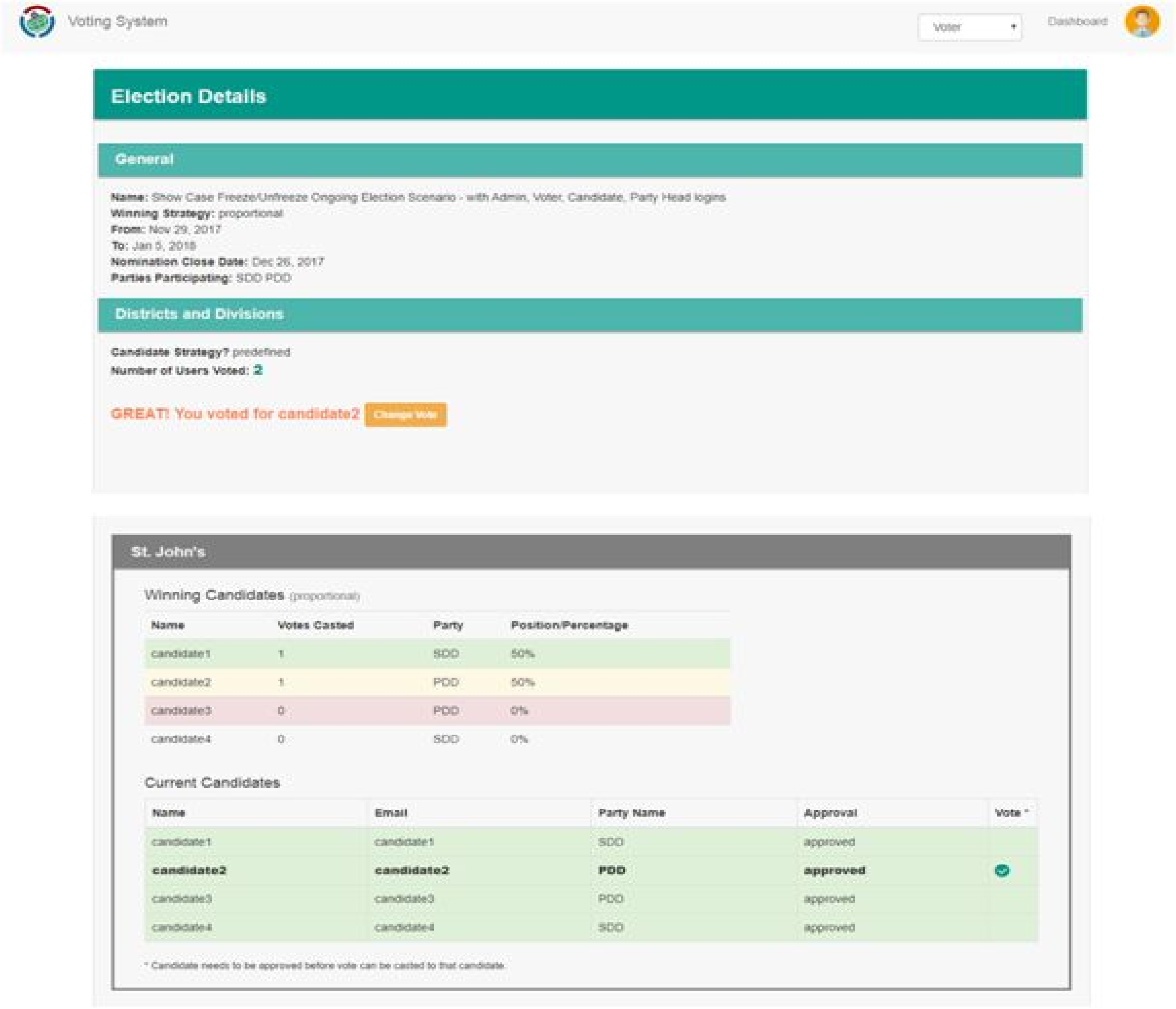
Nov 28, 2017 — Nov 28, 2017

Show case - Winner Takes All - Tie Breaker - Round 2 - Show sequence of previous rounds with link

Round: 2

Winning Strategy
winnerTakesAll
Candidates strategy
nomination

Nov 23, 2017 — Nov 24, 2017



Election Details

General

Name: Show Case Freeze/Unfreeze Ongoing Election Scenario - with Admin, Voter, Candidate, Party Head logins
Winning Strategy: proportional
From: Nov 29, 2017
To: Jan 5, 2018
Nomination Close Date: Dec 26, 2017
Parties Participating: SDD PDD

Districts and Divisions

Candidate Strategy? predefined
Number of Users Voted: 2

GREAT! You voted for candidate2

St. John's

Winning Candidates (proportional)

Name	Votes Casted	Party	Position/Percentage
candidate1	1	SDD	50%
candidate2	1	PDD	50%
candidate3	0	PDD	0%
candidate4	0	SDD	0%

Current Candidates

Name	Email	Party Name	Approval	Vote *
candidate1	candidate1	SDD	approved	
candidate2	**candidate2**	**PDD**	**approved**	✓
candidate3	candidate3	PDD	approved	
candidate4	candidate4	SDD	approved	

* Candidate needs to be approved before vote can be casted to that candidate.

URLs

Website URL

http://sc-4.cs.mun.ca/#/

Github URL

https://github.com/harisbarki/CS6905-Online-Voting